FEKO
User's Manual

Suite 5.1

December 2005

Copyright 1998 – 2005: **EM Software & Systems-S.A. (Pty) Ltd**
32 Techno Avenue, Technopark, Stellenbosch, 7600, South Africa
Tel: +27-21-880-1880, Fax: +27-21-880-1936
E-Mail: feko@emss.co.za
WWW: http://www.feko.info

Contents

1 Introduction **1-1**

1.1 FEKO Overview 1-1

1.2 Contacting your distributor or EMSS 1-3

2 General comments **2-1**

2.1 Structure of the input file 2-1

2.2 Modelling and meshing guidelines 2-2

2.2.1 Definitions and terms 2-2

2.2.2 Meshing guidelines regarding element sizes 2-3

2.2.3 Meshing guidelines regarding connectivity 2-5

2.3 Usage and concept of labels 2-8

2.4 Utilisation of symmetry 2-10

2.4.1 Geometric symmetry 2-10

2.4.2 Electric symmetry 2-11

2.4.3 Magnetic symmetry 2-11

2.4.4 Example of the application of symmetry 2-12

2.4.5 Special enforcement of symmetry: Even – odd method 2-13

2.5 Dielectric solids 2-14

2.6 Dynamic memory management 2-16

2.6.1 Telling FEKO how much memory can be used 2-16

2.6.2 Other variables that are under user control 2-17

2.6.3 Variables that are automatically set correctly 2-17

2.7 Summary of the files 2-19

2.8 Environment variables 2-22

2.9 Checking the validity of the results 2-26

3 The licence manager **3-1**

3.1 Displaying licence information 3-1

3.2 Managing floating licences 3-2

3.2.1 Preferred licence 3-3

3.2.2 Managing the server 3-4

3.3 Determine the machine code 3-4

4 Working in CADFEKO **4-1**

4.1 Introduction 4-1

4.1.1 Starting CADFEKO 4-1

4.1.2 Display settings and graphics cards 4-1

4.2 CADFEKO overview 4-3

4.2.1 The standard toolbar 4-4

4.2.2 Creating models 4-5

4.3 The 3D view 4-5

4.3.1 The workplane 4-6

4.3.2 Transforming the workplane 4-8

4.3.3 Snap mode 4-8

4.3.4 The 3D view toolbar 4-9

4.3.5 Cutplanes 4-10

4.3.6 Selective viewing 4-10

4.4 The tree 4-11

4.5 Variables and named points 4-13

4.5.1 Named points 4-15

4.5.2 Calculator 4-15

4.6 Creating geometry 4-16

4.6.1 Geometry extents 4-16

4.6.2 Local coordinates 4-16

4.6.3 Point entry 4-17

4.6.4 Creating solid primitives 4-18
4.6.5 Creating surface primitives 4-19
4.6.6 Creating curve primitives 4-21
4.6.7 Importing / exporting geometry 4-21
4.7 Operations on geometry 4-23
4.7.1 Transformations 4-24
4.7.2 Multiple (transformed) copies 4-24
4.7.3 Boolean operations 4-26
4.7.4 Split parts 4-26
4.7.5 Spinning and sweeping parts 4-26
4.7.6 Loft surfaces 4-27
4.7.7 Projection and imprinting points 4-28
4.7.8 Reversing face normals 4-29
4.7.9 Removing detail and the simplify operation 4-29
4.7.10 Copy objects 4-30
4.7.11 Copy original 4-31
4.7.12 Exploding parts 4-31
4.8 Specifying dielectric media 4-31
4.8.1 Metal faces 4-33
4.9 Geometry validation 4-33
4.10 Assemblies 4-34
4.11 Selecting items 4-34
4.12 Creating meshes 4-36
4.12.1 Specifying mesh parameters 4-38
4.12.2 Importing meshes 4-39
4.13 Mesh validation and editing 4-40
4.13.1 Mesh information 4-40
4.13.2 Relabelling mesh elements 4-41
4.13.3 Transforming mesh parts 4-41

4.13.4 Merging coincident vertices 4-41
4.13.5 Merging elements (removing vertices) 4-42
4.13.6 Remove duplicate and collapsed elements 4-42
4.13.7 Finding distorted and oversized elements 4-43
4.13.8 View free mesh edges and segment nodes 4-43
4.13.9 Editing mesh vertices 4-44
4.13.10 Creating mesh triangles 4-44
4.13.11 Reversing normals 4-44
4.13.12 Setting mesh properties 4-45
4.14 Preferences 4-46
4.15 Running components 4-46
4.15.1 Solution options 4-46
4.16 Working with CADFEKO models in EDITFEKO 4-46
4.16.1 Setting units 4-47
4.16.2 Referencing elements 4-47
4.16.3 Using variables and named points in EDITFEKO 4-47
4.16.4 Setting dielectric parameters 4-47
4.16.5 Using symmetry 4-48
4.17 Notes editor 4-48
4.18 Getting help 4-49
4.19 Short-cut keys 4-49

5 The editor EDITFEKO **5-1**
5.1 Toolbars 5-2
5.1.1 File control toolbar 5-2
5.1.2 The edit and window toolbars 5-3
5.2 Menu items in EDITFEKO 5-3
5.2.1 File menu 5-3
5.2.2 Edit menu 5-3

5.2.3 Geometry cards and Control cards menus 5-4
5.2.4 Run menu 5-4
5.2.5 Window menu 5-5
5.2.6 Help menu 5-5
5.3 Preferences in EDITFEKO 5-5
5.3.1 Appearance tab 5-6
5.3.2 Options tab 5-6
5.4 PREFEKO mode 5-6
5.4.1 Generating input cards 5-6
5.4.2 Variable editor 5-9
5.5 OPTFEKO mode 5-9
5.5.1 Parameter tab 5-9
5.5.2 Penalty function tab 5-10
5.5.3 Optimisation tab 5-10
5.5.4 Aim function tab 5-10
5.6 Important keystrokes 5-10

6 Working in POSTFEKO 6-1
6.1 Introduction 6-1
6.1.1 Starting POSTFEKO 6-1
6.2 POSTFEKO overview 6-2
6.3 Displaying the geometry and model validation 6-4
6.3.1 Zooming / rotating / panning the model 6-4
6.3.2 The model view control panel 6-5
6.4 Displaying 3D results 6-7
6.4.1 Result view toolbar 6-8
6.4.2 3D result selection 6-9
6.5 Displaying 2D results 6-10
6.5.1 2D result selection 6-10

6.5.2 Continuous (adaptive) frequency sampling 6-11
6.5.3 New 2D graph toolbar 6-11
6.5.4 Series manipulation toolbar 6-12
6.5.5 Graph settings toolbar 6-13
6.5.6 Zooming / rotating / panning the model 6-14
6.5.7 Value series description 6-14
6.5.8 PFG graphs and importing GraphFEKO WFG graphs 6-15
6.5.9 Loading and saving raw data 6-15
6.6 Exporting images 3D / 2D 6-16
6.7 Polarisation 6-16
6.8 Viewing the output file 6-17
6.9 Advanced / troubleshooting features 6-17
6.9.1 Database browser 6-17
6.9.2 Help menu 6-18

7 The preprocessor PREFEKO 7-1
7.1 Description 7-1
7.2 Running PREFEKO 7-1
7.3 Symbolic variables 7-2
7.4 FOR/NEXT loops 7-7
7.5 IF/ELSE/ENDIF constructs 7-9
7.6 Symbolic node names 7-10
7.7 PRINT and EXIT commands 7-11

8 The FEKO solution kernel 8-1
8.1 Introduction 8-1
8.2 Running the FEKO kernel 8-1
8.2.1 Running the sequential version 8-1
8.2.2 Running the parallel version 8-2
8.2.3 Running on a remote host 8-5

9 Description of the geometry cards 9-1

9.1 Overview of the geometry cards 9-1

9.2 Alphabetical description of the geometry cards 9-3

9.2.1 ** card 9-3

9.2.2 BL card 9-4

9.2.3 BP card 9-6

9.2.4 BQ card 9-8

9.2.5 BT card 9-10

9.2.6 CB card 9-12

9.2.7 CL card 9-14

9.2.8 CN card 9-16

9.2.9 DK card 9-17

9.2.10 DP card 9-19

9.2.11 DZ card 9-20

9.2.12 EG card 9-22

9.2.13 EL card 9-25

9.2.14 FM card 9-27

9.2.15 FO card 9-28

9.2.16 FP card 9-30

9.2.17 HE card 9-31

9.2.18 IN card 9-33

9.2.19 IP card 9-52

9.2.20 KA card 9-53

9.2.21 KK card 9-54

9.2.22 KL card 9-58

9.2.23 KR card 9-59

9.2.24 KU card 9-62

9.2.25 LA card 9-64

9.2.26 ME card 9-65

9.2.27 NU card 9-68
9.2.28 PB card 9-70
9.2.29 PH card 9-72
9.2.30 PM card 9-74
9.2.31 PO card 9-76
9.2.32 PY card 9-80
9.2.33 QT card 9-81
9.2.34 QU card 9-82
9.2.35 RM card 9-84
9.2.36 SF card 9-88
9.2.37 SY card 9-90
9.2.38 TG card 9-91
9.2.39 TO card 9-94
9.2.40 TP card 9-96
9.2.41 UT card 9-98
9.2.42 UZ card 9-100
9.2.43 VS card 9-101
9.2.44 WG card 9-104
9.2.45 ZY card 9-105

10 Description of the control cards **10-1**

10.1 Overview of control cards and execution sequence 10-1
10.2 Detailed description of the control cards 10-5
10.2.1 ** card 10-5
10.2.2 Ax Cards 10-6
10.2.3 A0 card 10-9
10.2.4 A1 card 10-12
10.2.5 A2 card 10-13
10.2.6 A3 card 10-14

10.2.7 A4 card 10-16
10.2.8 A5 card 10-18
10.2.9 A6 card 10-19
10.2.10 A7 card 10-21
10.2.11 AC card 10-23
10.2.12 AE card 10-26
10.2.13 AI card 10-29
10.2.14 AP card 10-31
10.2.15 AR card 10-38
10.2.16 AS card 10-43
10.2.17 AV card 10-49
10.2.18 AW card 10-51
10.2.19 BO card 10-54
10.2.20 CF card 10-56
10.2.21 CG card 10-58
10.2.22 CM card 10-62
10.2.23 CO card 10-63
10.2.24 DA card 10-66
10.2.25 DI card 10-69
10.2.26 EN card 10-70
10.2.27 FE card 10-71
10.2.28 FF card 10-79
10.2.29 FR card 10-83
10.2.30 GF card 10-86
10.2.31 L4 card 10-92
10.2.32 LD card 10-94
10.2.33 LE card 10-95
10.2.34 LP card 10-97
10.2.35 LS card 10-98

10.2.36 LZ card 10-99
10.2.37 OF card 10-100
10.2.38 OS card 10-101
10.2.39 PS card 10-103
10.2.40 PW card 10-105
10.2.41 SA card 10-109
10.2.42 SK card 10-111
10.2.43 SP card 10-117
10.2.44 TL card 10-118

11 The optimiser OPTFEKO **11-1**
11.1 Preparing input for OPTFEKO 11-1
11.1.1 More on the *.pre file 11-1
11.1.2 More on the *.opt input file 11-1
11.2 Using EDITFEKO to create a *.opt file 11-2
11.2.1 The optimising toolbar 11-2
11.2.2 Definition of optimisation parameters 11-3
11.2.3 Definition of the penalty function 11-4
11.2.4 Definition of the optimisation process 11-5
11.2.5 Defining the aim function 11-12
11.3 Running OPTFEKO 11-27
11.4 An example using OPTFEKO 11-28

12 The program TIMEFEKO **12-1**
12.1 Description 12-1
12.2 The *.pre input file 12-1
12.3 The *.tim input file 12-2
12.3.1 Defining the pulse form 12-2
12.3.2 Definition of the frequency block 12-6
12.3.3 Definitions of the normalisation 12-7

12.3.4 Definition of the excitation output 12-7
12.3.5 Definition of a time point 12-7
12.4 Running TIMEFEKO 12-8
12.5 TIMEFEKO output 12-8
12.6 A TIMEFEKO example 12-9

13 The program ADAPTFEKO **13-1**
13.1 Description 13-1
13.2 Running ADAPTFEKO 13-1
13.3 The *.pre input file 13-1
13.4 ADAPTFEKO example 13-2

14 Description of the output file of FEKO **14-1**
14.1 Geometric data 14-1
14.2 Excitation 14-7
14.3 Currents and charges 14-8
14.4 Finite conductivity 14-9
14.5 Near fields 14-10
14.6 Far fields 14-11
14.7 S-parameters 14-15
14.8 Computation time and peak memory 14-16

15 Copyright notices and acknowledgements **15-1**
15.1 Copyright to Voronoi 15-1
15.2 Copyright to SuperLU 15-1
15.3 Copyright of libcurl 15-2
15.4 Qwt project usage 15-3
15.5 HOOPS and Parasolid 15-3
15.6 MeshSim 15-3

Index **I-1**

1 Introduction

The name FEKO is an abbreviation derived from the German phrase **FE***ldberechnung bei* **K***örpern mit beliebiger* **O***berfläche.* (*Field computations involving bodies of arbitrary shape.*) As the name suggests FEKO can be used for various types of electromagnetic field analyses involving objects of arbitrary shapes.

Changes in this manual with respect to the previous one of July 2005 (Suite 5.0) are indicated as follows:
Sections that have changed from those in the previous version of the manual.
Sections that were newly added to this version of the manual.

1.1 FEKO Overview

The core of the program FEKO is based on the *Method of Moments* (MoM). Electromagnetic fields are obtained by first calculating the electric surface currents on conducting surfaces and equivalent electric and magnetic surface currents on the surface of a dielectric solid. The currents are calculated using a linear combination of basis functions, where the coefficients are obtained by solving a system of linear equations. Once the current distribution is known, further parameters can be obtained e.g. the near field, the far field, radar cross sections, directivity or the input impedance of antennas.

Electrically large problems are usually solved with either the Physical Optics (PO) approximation and its extensions or the Uniform Theory of Diffraction (UTD). In FEKO these formulations are hybridised with the MoM at the level of the interaction matrix[1]. This is a major step in addressing the problem of solving electromagnetic problems where the object under consideration is too large (in terms of wavelengths) to solve with the MoM but too small to apply only the asymptotic UTD approximation with high accuracy. With the hybrid MoM/PO or hybrid MoM/UTD techniques, critical regions of the structure can be considered using the MoM and the remaining regions (usually larger, flat or curved metallic surfaces) using the PO approximation or UTD.

In addition to these PO or UTD high frequency techniques, since FEKO Suite 4.2 also a fast integral equation solver, the Multilevel Fast Multipole Method (MLFMM) has been included. As compared to the MoM, this drastically reduces the memory requirement and CPU-time, and as compared to the PO / UTD asymptotic techniques, it is a rigorous technique not based on specific high frequency approximations (which in the case of UTD are for instance only available for certain canonical objects limiting the range of applicability).

For the modelling of dielectric/magnetic bodies, the MoM as implemented in FEKO offers a number of different techniques such as the surface equivalence principle, the

[1]The UTD implementation is currently limited to an arbitrary number of perfectly conducting, flat, polygonal surfaces or one perfectly conducting circular cylinder.

volume equivalence principle, special Green's functions for planar multilayer media, or also approximations for coatings or thin dielectric sheets. Since FEKO Suite 5.0, these methods are complemented by the *Finite Element Method* (FEM) which is available in a MoM/FEM hybrid formulation, and well suited to model highly heterogeneous objects. But also for homogeneous dielectric bodies the FEM might require less memory than the MoM.

Only time domain harmonic sources are supported in the current version, and consequently calculation is done in the frequency domain. FEKO uses the $e^{j\omega t}$ time convention. Different sources are available including an incident plane wave, various voltage gap formulations (on wire segments as well as triangle edges), a magnetic ring current (TEM-frill, with which a coaxial feed can be modelled) and impressed currents and patterns.

Regarding the FEKO user interface, this consists of three main components CADFEKO, EDITFEKO, and POSTFEKO. CADFEKO is used to graphically create a geometry and do the required meshing for the FEKO solution kernel. Alternatively, the geometry can be defined in terms of high level commands in an input file (`*.pre`) which also sets the solution parameters. The customised text-editor EDITFEKO assists the user in creating and editing the input file. The preprocessor/mesher PREFEKO processes this file and prepares the input file (`*.fek`) for the program FEKO which is the actual field calculation code.

PREFEKO also supports the direct import of *meshed* geometry in FEMAP neutral, NASTRAN, PATRAN, AutoCAD `*.dxf`, STL, or ANSYS `*.cdb` formats. PREFEKO can also import and mesh polyline surfaces from unmeshed `*.dxf` files.

The ASCII (`*.out`) and binary (`*.bof`) output files of FEKO contain all the solution information. The resulting fields and/or currents can be displayed in 3D or 2D with POSTFEKO. POSTFEKO is used not only for post processing purposes, but also to visualise the geometry of the FEKO model, with excitations, near-field points etc. before the actual FEKO run.

Note that the FEKO kernel components are available on PC's and a wide variety of workstations. The GUI components CADFEKO, EDITFEKO, and POSTFEKO are available on PC's running MS Windows or Linux only. All pre- and post processing must thus be performed on a PC, while the actual computationally intensive field calculations can be performed on the PC itself, or on a workstation or parallel cluster as required. FEKO includes a remote launching facility to make such a remote execution easy to use from within the GUI running on the PC.

First time users are advised to work through the *Demo example* and the *Getting started manual* (located in the `doc` directory under the FEKO home directory — the directory where FEKO has been installed). It gives a basic introduction to FEKO and the different FEKO modules. It is also recommended that new users read the "General Comments" (Chapter 2) of this *User's manual* carefully.

Various simple FEKO examples that show the application of the different cards are discussed in the *Examples guide.*

1.2 Contacting your distributor or EMSS

You can find the distributor for your region at

www.feko.info/contact.htm

Alternatively, for technical questions, please send email to

feko_support@emssusa.com	for North America
feko_support@emss.de	for Europe
feko_support@emss.co.za	for all other regions

or, for activation codes and licence queries, to

feko_license@emssusa.com	for North America
feko_license@emss.de	for Europe
feko_license@emss.co.za	for all other regions

2 General comments

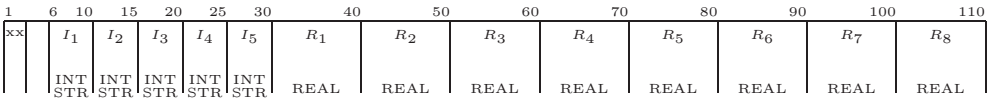
2.1 Structure of the input file

The main FEKO kernel is using a card based `*.pre` file as input, similar to the thin wire MoM code NEC. Such an input file consists of various cards (high level commands) which can be categorised into geometry cards and control cards. The geometry and the field parameters to be calculated are specified using these cards. Ons of these cards allows importing meshed geometry. This is used to integrate CADFEKO models into the final FEKO input file. Certain geometric cards are interpreted and filtered out by the preprocessor PREFEKO and translated into other cards.

Normally users are advised to use EDITFEKO, then there are panels where the various settings can be made, and also the format of how each card is written back to the main file edit window will be correct.

The card format is described here mainly for users wanting to create a FEKO input file automatically by some other code. FEKO supports two different formats for cards, the colon separated and the column based format. These formats can also be mixed in an input file.

- **Column based format:** The basic form of the input cards is



  The upper numbers indicate the columns. The name field in columns 1 and 2 specifies the type of the card. This is followed by five integer parameters I_1 to I_5 (when read with PREFEKO these input fields may also contain strings such as node names) containing five digits each, and eight real parameters R_1 to R_8 containing ten digits each.

- **Colon separated format:** This is some kind of free format input where the individual integer and real parameters are separated by the colon character. The main advantage is that then integer and real input fields are not restricted to 5 or 10 characters, respectively. For this card format, still in columns 1 and 2 the card name must be used, then followed immediately in column 3 by a colon. The rest is free format, so for instance this is a valid syntax:

```
DP: S1 :  :  :  : #x : #y : #z
BP: S2 : S2 : S3 : S4
```

All input and output parameters of FEKO are in SI units, e.g. lengths are in metres, potential in volts, etc. (Note the angles are in degrees.) FEKO includes various scaling options (see the SF, TG and IN cards) such that dimensions may be entered in different units and scaled to metres.

The principal structure of the input file is shown below:

`**`	*Comments at the start of the input file*
...	*Cards that define the geometry*
`EG`	*End of the geometry*
...	*control cards that define the excitation and indicate which field quantities shall be calculated*
`EN`	*End of the input file*

Chapter 9 gives an overview of the geometric cards with detailed descriptions of the individual cards (also how to import external meshes or CADFEKO models using the IN card). Similarly Chapter 10 gives an overview and description of the control cards.

2.2 Modelling and meshing guidelines

2.2.1 Definitions and terms

Conducting surfaces are subdivided into triangles, and wires into segments. For dielectrics, there are a number of possibilities — see the discussion "Dielectric Solids" in section 2.5. Using the surface current method the surface of the dielectric solid is also subdivided into triangles, whereas with the volume current method, the solid is subdivided into cuboids. For the FEM, the mesh is based on tetrahedral volume elements.

Given below are a number of definitions that are used frequently in this manual:

Segment: A short section of a wire (short in comparison with the wavelength).

Node: The point where two segments are joined is called a node. One basis function is assigned to each node.

Edge: The common line between two adjacent triangles. If the surface is a metal, then one basis function is assigned to each edge. If the surface is a dielectric, then two basis functions are assigned for the current density, one for the equivalent electric current density and one for the equivalent magnetic current density.

Connection point: A connection point is where a segment is joined to a triangle. The end of the segment is connected to the vertex of the triangle. A basis function is assigned to each connection point.

Cuboid: A volume element used to model dielectric and magnetic solids according to the volume current method in the MoM. It has 90 degree corners similar to a cube, but does not need to have equal side lengths.

Tetrahedron: 3D tetrahedral shaped volume element used in conjunction with the FEM.

Polygon: A planar surface element with straight edge boundaries.

2.2.2 Meshing guidelines regarding element sizes

For the meshing process the following rules must be adhered to:

- The segment length l should be smaller than $\frac{\lambda}{10}$, where λ is the free space wavelength.
- Note also that the segment current flows only in the axial direction. Thus segments should not be too short relative to the wire radius. Ideally the segment length should be at least four times the radius.
- The area A of each triangular element should be smaller than $\frac{\lambda^2}{70}$. For an equilateral triangle with side length s, the area is given by $A = \frac{\sqrt{3}}{4} s^2$. Using this formula, it is found that the edge length s should be shorter than approximately $\frac{\lambda}{5...6}$. According to the geometry and the need for accuracy, more or less triangles may be needed. If the memory constraints allow it, a segment length of $\frac{\lambda}{8...10}$ is preferred.
- When modelling a surface by means of a wire grid, the radius should be chosen so that the wire area in one direction is approximately the same as the area of the surface to be modelled as a wire grid. From the approximation $2\pi r \cdot l \approx l^2$, one finds the wire radius to be
$$r \approx \frac{l}{2\pi},$$
where $l \approx \frac{\lambda}{10}$ defines the segment length.
- The length on dielectric cuboids has to be small in comparison with the wavelength λ *in the dielectric* as well as the skin depth
$$\delta = \sqrt{\frac{2}{\omega\mu\sigma}}.$$
Due to the staircase approximation used, a mesh size of $\frac{\min(\lambda,\delta)}{10}$ or finer is recommended.
- When meshing the FEM region into tetrahedral volume elements, then as a guideline an element size (edge length of the tetrahedra) of $\frac{\lambda}{5}$ should be used inside the FEM volume, but for those elements right on the FEM/MoM interface, a finer element

size of $\frac{\lambda}{10}$ is recommended. The reason why inside a coarser mesh can be used is that higher order basis functions are employed there. It should be noted that λ represents the wavelength *inside of the dielectric* medium.

As a trick, to reduce the overall memory requirement of FEKO, it might be advantageous to add a small air region buffer around the actual dielectric object and also mesh this into tetrahedral elements (i.e. also use the FEM there). Then since in air the wavelength is larger one can use larger tetrahedral elements at the FEM/MoM interface, which reduces the memory requirement of the FEM/MoM hybrid method for the coupling arrays. This reduction is typically much higher than the extra memory required due to having more tetrahedral elements (mesh of the extra air buffer zone).

- In some cases accurate modelling of the geometry requires significantly finer mesh elements than specified by the guidelines above. (For low frequencies in particular, the segmentation rule of $\frac{\lambda}{10}$ is often much too coarse to yield a reasonable representation of the geometry.) Another case where finer discretisation may be required is where a wire runs parallel to a conducting plate. If the wire is closer than $\frac{\lambda}{10}$ to the plate, the size of the triangles in the direction orthogonal to the wire should be similar to the distance from the wire to the plate in order to give an accurate representation of the surface charge distribution.

If the segmentation rules are not adhered to, then the following errors and warnings will be reported by the FEKO kernel:

	Warning	Error
Ratio of the segment length to the wavelength	$\frac{l}{\lambda} > 0.3$	$\frac{l}{\lambda} > 0.5$
Ratio of the segment radius to the segment length	$\frac{\rho}{l} > 0.3$	$\frac{\rho}{l} > 1.0$
Ratio of the area of the triangles to the wavelength squared	$\frac{A}{\lambda^2} > \frac{1}{30}$	$\frac{A}{\lambda^2} > \frac{1}{10}$
Ratio of the cuboid edge length to the wavelength	$\frac{l}{\lambda} > 0.25$	$\frac{l}{\lambda} > 0.5$
Ratio of the cuboid edge length to the skin depth	$\frac{l}{\delta} > \frac{1}{5}$	$\frac{l}{\delta} > \frac{1}{3}$
Ratio of the tetrahedral face area to the wavelength squared (inner mesh elements)	$\frac{A}{\lambda^2} > 0.047$	$\frac{A}{\lambda^2} > 0.433$
Ratio of the tetrahedral face area to the wavelength squared (boundary surface mesh elements)	$\frac{A}{\lambda^2} > 0.033$	$\frac{A}{\lambda^2} > 0.108$

2.2.3 Meshing guidelines regarding connectivity

The segmentation (or meshing) is performed automatically by CADFEKO or PREFEKO (or also a combination of the two for mixed models or if for instance mesh refinement is used with the RM card).

There are some rules that must be adhered to for the final mesh. Most of these rules are automatically complied with when creating FEKO models in CADFEKO. However, users must take care of these rules when for instance combining CADFEKO and PREFEKO models (e.g. attaching an antenna modelled with geometry cards on an aircraft meshed in CADFEKO) or when creating the geometry solely in PREFEKO.

Wires can only be connected at the end points of the respective segments. They are not allowed to overlap. An example is shown in figure 2-1. When using geometry cards, then the wires AB and CD are not allowed to be entered in this way. AB will be subdivided into segments so that point C is not recognised as an ohmic connection. This is shown, on the left side, in figure 2-2. The line AB is divided into four segments and the point C is not on a node. To resolve this problem three wires have to be defined AC, CB and CD. Then there will be an ohmic connection at point C, as shown in figure 2-2 on the right.

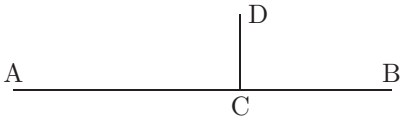


Figure 2-1: Example of a wire structure

Figure 2-2: Incorrect (left) and correct (right) subdivision into segments

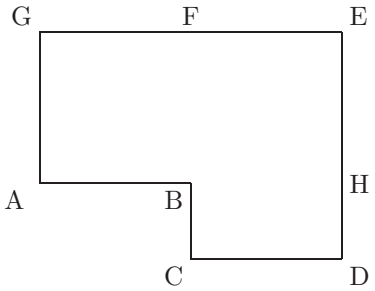


Figure 2-3: Example for a surface

A similar rule has to be followed when surfaces are subdivided into triangles. If the surface in figure 2-3 has to be meshed, there are a number of possibilities. The surface can be subdivided into the rectangles ABFG and CDEF. This can result in the mesh shown in figure 2-4 (on the left-hand side). In this case, there is no ohmic connection at the line BF, because the triangles' vertices are not connected (in the sense that the triangles do not have common edges).

The correct subdivision is shown in figure 2-4 (in the middle). Here the rectangles ABFG, CDHB and BHEF were used. Another subdivision is possible using quadrangles ABEG, and BCDE (figure 2-4, right side). BE is now a common edge and the surface will be meshed correctly. Care has to be taken to ensure that when two surfaces touch, the common edge is part of both surfaces. In figure 2-3 the edge BF is such an edge. That is why the right part of the surface is not allowed to be defined as the rectangle CDEF, because BF is not an edge of this rectangle. The recent introduction of the PM card now also allows defining a single polygon ABCD(H)E(F)G resulting in a correct mesh. Also when using CADFEKO and meshing a polygonal structure with an arbitrary number of corner points, the created mesh will be correct.

A connection point between a segment and one or more triangles is only recognised when the beginning or the end of the segment lies on the vertex or vertices of the triangles. In figure 2-5 the left side is an incorrect and the right side a correct connection (here the segment is connected to six triangles).

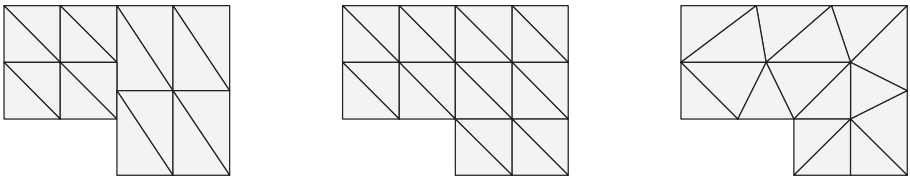


Figure 2-4: Incorrect (left) and correct (middle and right) subdivision into triangles

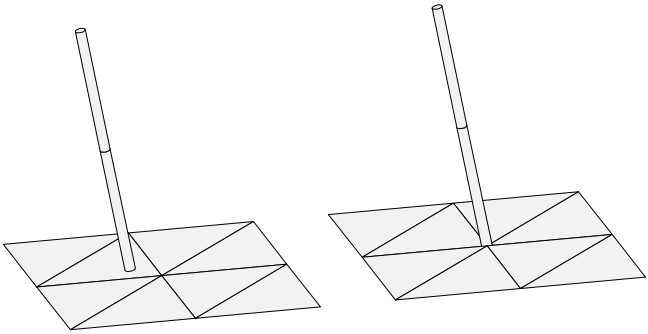


Figure 2-5: Incorrect (left) and correct (right) connection between a segment and triangles

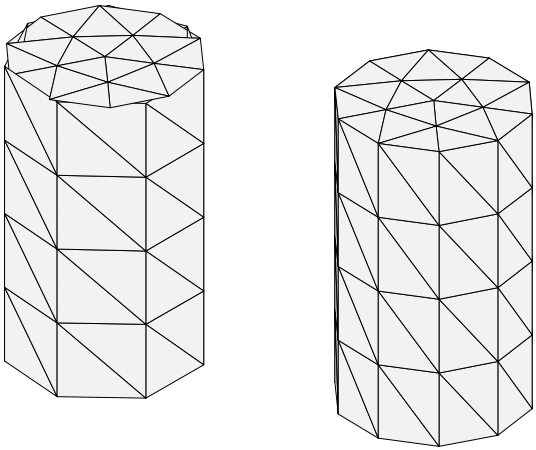


Figure 2-6: Incorrect (left) and correct (right) connection between curved edges

When curved structures (circles, spheres) are modelled, a finer mesh may be used along the curved edges to get a more accurate representation of the geometry. When connecting curved edges in PREFEKO, the same maximum edge length has to be used for both edges. See the example in figure 2-6. Again, also here CADFEKO does this proper meshing automatically.

When meshing dielectric solids into tetrahedral elements for the FEM, then the faces of adjacent tetrahedra must match, otherwise there would be a gap. Also when modelling for instance a patch antenna and using metallic triangles to model the patch and tetrahedral elements to represent the dielectric substrate, then also the metallic triangular elements must match the triangular facets of the tetrahedral volume elements. See the example in figure 2-7 where this is illustrated.

In order to ensure proper connectivity of elements, the FEKO kernel has several checks built in and will give errors if e.g. overlapping triangular elements are found. Also CADFEKO allows the user to perform several checks, see section 4.9.

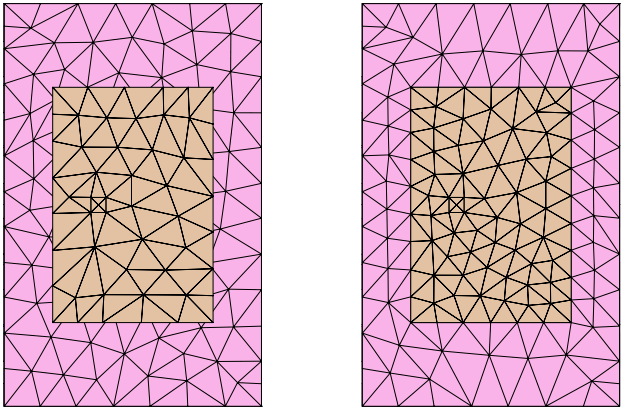


Figure 2-7: Incorrect (left) and correct (right) modelling of metallic surfaces at the boundary of a FEM region

2.3 Usage and concept of labels

In FEKO labels are an important concept to set electromagnetic properties or to do selections using only part of the geometry. For instance in figure 2-8 the label `Feed` can be used to define the feed segment (voltage source), or the label `MainRefl` can be used to instruct FEKO to use physical optics for the main reflector.

Labels are set either directly in CADFEKO, or when using geometry cards then in EDITFEKO using the LA card (see section 9.2.26). When importing certain mesh formats (see IN card in section 9.2.18) then also labels can be imported (for instance the NASTRAN property gets converted into a FEKO label). In principle, FEKO labels can be:

- A positive integer number (including zero), then also valid expressions like `3*#i+2` can be used which are evaluated.
- A string of characters (valid are 'a'-'z', 'A'-'Z', '0'-'9' and the underscore '_'), optionally followed by a variable (which starts with the '#' sign). Such variables at the end are evaluated and replaced by the corresponding numerical value (rounded to integer). Note that string labels are case insensitive in FEKO, i.e. labels `Roof` and `ROOF` are treated identically inside FEKO.

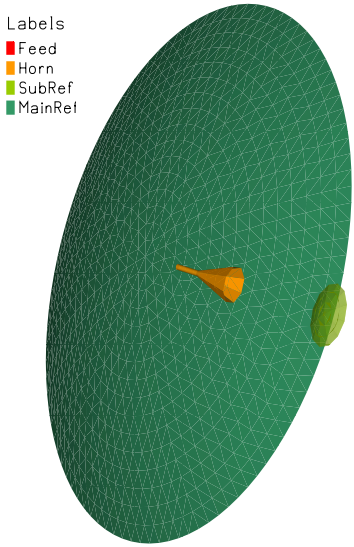


Figure 2-8: Example demonstrating the usage of labels (display of labels in colour with legend in POSTFEKO)

So for instance these labels are valid

```
23
5*#k+#j/2
LeftWing
Front_Door
Part#i
```

while these labels are not valid

```
Left+Wing    (invalid character '+')
-23          (negative integer)
Part_#i_#k   (two variables)
```

With the CB card, FEKO allows to change labels (e.g. after having imported some geometry). A powerful wildcard globbing is supported there, see section 9.2.6 for details.

At certain FEKO cards one can specify ranges of labels, or also sometimes labels are incremented (e.g. when using symmetry). The algorithm for this is that after having evaluated expressions or having replaced variables, a label is split into the associated number and the remaining base string. The associated number is split off from the back of the label, and if there are no digits, this is set to zero. Examples:

Label name	Base string	Associated number
1		1
Roof	Roof	0
Part_17	Part_	17

When incrementing labels, the base string is kept and the associated number is incremented. There is just one exception, the label zero will always remain zero. So for instance when incrementing labels by 2 we get this:

Label name	Incremented label
1	3
Roof	Roof
Part_17	Part_19

When using ranges of labels, then for the range specification to be valid the base strings must match, and the associated number must be in the correct range between the associated numbers of start and end label. So for instance the label `Part_12` is in the range of the labels `Part_10` ... `Part_20`, but the label `Part_5` is outside this range. Using a range like `Front` ... `Back` is not valid (different base strings).

2.4 Utilisation of symmetry

It is possible to reduce the calculation time and memory usage if symmetry is utilised. This can be done by using the SY card (section 9.2.37).

Three coordinate planes, $x = 0$ (yz plane), $y = 0$ (xz plane) and $z = 0$ (xy plane) may be defined as planes of symmetry. There are three different types of symmetry. They are described below.

2.4.1 Geometric symmetry

With this type of symmetry the geometry of the modelled solid or part of the solid is symmetric about one or more coordinate planes. *The interaction between any two basis functions must be the same as that between their symmetrical counterparts. Everything which affects this must be symmetrical, i.e. loading, losses, Green's functions, etc.* The source, however, is not symmetric, thus a symmetric current distribution does not exist. This asymmetric current distribution leads to asymmetric electric and magnetic fields.

The body of a truck with an antenna placed at the front left hand side, will be used as an example. In the input file, half of the body is constructed (either the left or right side).

The other half is then created with the SY command. Finally the antenna is placed in the correct position on one side.

A rectangular metallic plate, illuminated by an electromagnetic wave from a direction outside the principle planes, is another example. In this case a quarter of the plate is constructed and the rest is created using the SY card (section 9.2.37) with geometric mirroring around two coordinate planes.

Geometric symmetry does not reduce the number of unknown coefficients in the current basis functions. Therefore there is no reduction in memory usage. There is, however, a reduction in computation time when the matrix elements are determined.

2.4.2 Electric symmetry

An electric symmetry plane is a plane which can be replaced by an ideal electrically conducting wall without changing the field distribution.

In figure 2-9 an electric symmetry plane is displayed. The electric $\vec{E}$ field's tangential component disappears and the magnetic $\vec{H}$ field's normal component disappears. The electric current density $\vec{J}$ is anti-symmetric and the magnetic current density $\vec{M}$ is symmetric.

As with geometric symmetry, less computational time is required to calculate the matrix elements. The number of unknown coefficients of the current basis functions are reduced, leading to linear equation system of a lower order. This leads to a further reduction in the computation time and requires less memory due to the reduction in matrix elements.

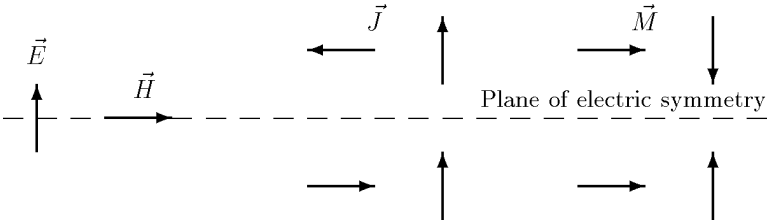


Figure 2-9: Electric symmetry plane

2.4.3 Magnetic symmetry

A magnetic symmetry plane is a plane which can be replaced by an ideal magnetically conducting wall without changing the field distribution.

In figure 2-10 a magnetic symmetry plane is displayed.

The electric $\vec{E}$ field's normal component and the magnetic $\vec{H}$ field's tangential disappears. The electric current density $\vec{J}$ is symmetric and the magnetic current density $\vec{M}$ is asymmetric.

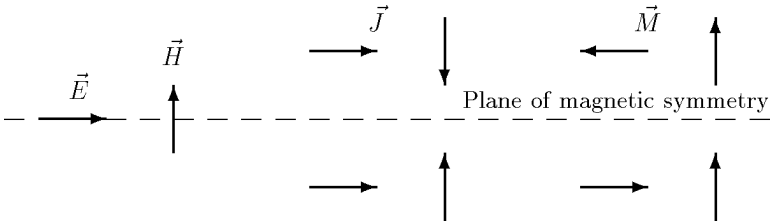


Figure 2-10: Magnetic symmetry plane

When using magnetic symmetry there is a reduction in the computational time when determining the matrix elements. The order of the matrix equation is reduced, which leads to a further reduction in the computational time and reduces the amount of memory needed to determine the matrix elements.

2.4.4 Example of the application of symmetry

In figure 2-11 a dielectric sphere is shown with a linear polarised incident electromagnetic field. The full description of the problem is given in `example_04` in the *Examples Guide*. Only the use of symmetry is described here.

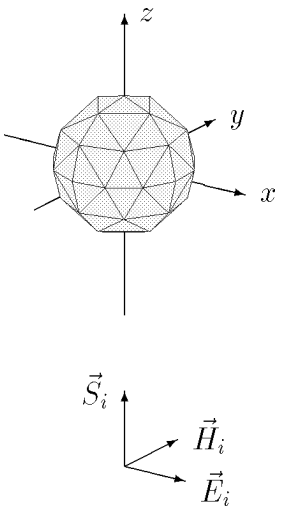


Figure 2-11: Dielectric sphere with incident field

The plane $z = 0$ (xy plane) is a plane of geometric symmetry, because the excitation does not have any symmetry in this plane. The plane $x = 0$ (yz plane) is a plane of electric symmetry, because the electric field is perpendicular to this plane and the magnetic

field has a tangential component only. Similarly the electric field only has a tangential component in the $y = 0$ plane and the magnetic field is perpendicular. This is thus a plane of magnetic symmetry.

To indicate the reduction in time and resources through the use of symmetry, a table is given below:

Type of Symmetry	Number of Unknowns	Memory usage	Solution Time in seconds
Symmetry not used	792	9.74 MByte	24.8
All 3 coordinate planes declared as planes of geometric symmetry	792	9.74 MByte	22.1
Plane $x = 0$ declared as electric plane of symmetry, plane $y = 0$ as magnetic plane of symmetry and plane $z = 0$ as geometric plane of symmetry	198	2.56 MByte	8.9

This example has relatively few unknowns. Most of the computational time is therefore used to determine the matrix elements in comparison to the time taken to solve the matrix equation. For applications with more unknowns, the reduction of unknowns could make a considerable difference in the time and memory required.

2.4.5 Special enforcement of symmetry: Even – odd method

The table in the previous section demonstrates the advantage of using electric and/or magnetic symmetry. For very large structures which have only geometrical symmetry, it may be worthwhile to consider two separate problems with electrical and magnetic symmetry as described below.

Figure 2-12 (a) shows the original problem. It consists of a dipole antenna with a passive wire below it. This is admittedly a very simple problem, normally this procedure would only be applied to much more complex structures.

The structure in figure 2-12 is symmetric about the plane $z = 0$, but the excitation is asymmetric and thus only geometric symmetry can be applied in FEKO. This problem may be separated into the two sub-problems shown in figures 2-12 (b) and (c). These two problems can be solved using respectively electric and magnetic symmetry about the plane $z = 0$. Each of these problems require only half the number of unknowns required for case (a). Superposition of the problems (b) and (c) — which must unfortunately be done manually as FEKO cannot do it automatically at the moment — yields the original problem.

The solution of each of the sub-problems requires only half of the storage space required for case (a). For very large problems the solution time is dominated by the time required

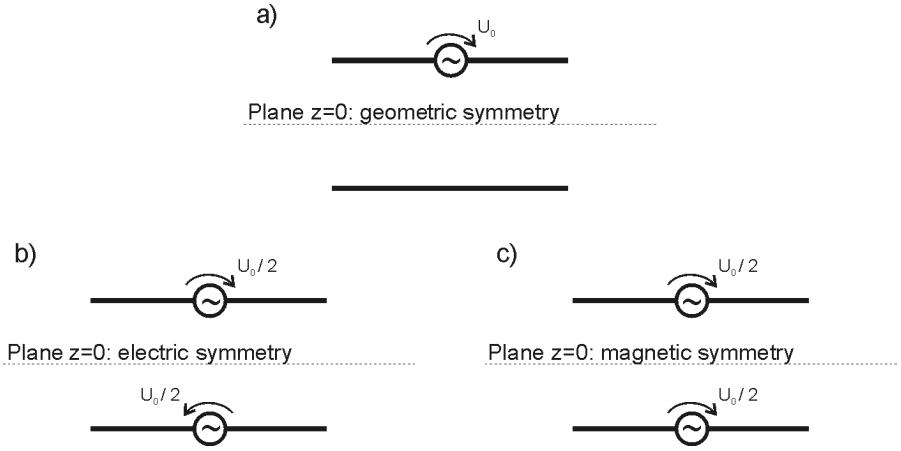


Figure 2-12: Separating a problem with geometrical symmetry into two problems with electric and magnetic symmetry respectively.

to solve the system of linear equations. In this case the two sub-problems each requires only one eight of the time required for case (a).

2.5 Dielectric solids

There are numerous ways to model dielectric objects in FEKO, three of these apply to arbitrarily shaped bodies:

- Method of Moments (MoM) with the surface equivalence principle. There interfaces between different homogeneous regions are subdivided into a surface mesh using triangular elements. Basis functions are applied to these elements for the equivalent electric and equivalent magnetic surface currents. Boundary conditions result through the use of equivalent sources.

  FEKO uses the normal vector of the triangles to distinguish the respective dielectric media on the two sides, see also the discussion at the ME card in section 9.2.26. It is advisable to check the normal vector orientation or directly the media usage (colouring by media) in POSTFEKO. `Example_04` (*Examples Guide*) shows a simple dielectric sphere. A more complex geometry, where a dielectric body is in contact with a conducting body, is described in `example_23`. Further details can be found there.

- MoM with the volume equivalence principle. Here the volume is subdivided into cuboidal elements. In principle, each element can be assigned a different material property. Inside the element the polarisation current is unknown. Normally since a

volume is meshed, many more unknowns as compared to a surface mesh result, so that this method requires a larger memory. However, this technique is very suitable for thin sheets and also very stable for low frequencies. Another advantage is that it is based on a so-called Fredholm integral equation of the second kind, making the formulation very stable, and thus when using MLFMM the number of iterations is typically small.

In `example_09` (*Examples Guide*) a dielectric cube is analysed with the volume current method.

- As alternative to the MoM, in FEKO also the Finite Element Method (FEM) is available. Like in the previous method, also 3D volumes are discretised, here into tetrahedral elements and not cuboidal ones. But as opposed to the MoM, the FEM matrices are sparse, thus the memory requirement for a FEM volume mesh is much less as compared to a MoM volume mesh of the same model.

  The FEM has also been hybridised with the MoM, see `example_42` or `example_43` (*Examples Guide*) for combined MoM/FEM examples.

For all these three techniques, the medium must either be set in CADFEKO, or through the use of ME cards (see section 9.2.26). By using the ME card it can be specified whether the triangular elements shall represent metallic surfaces or dielectric surfaces. For the surface equivalence principle, it is also possible to define metallic triangles and segments within the dielectric regions.

The material properties are assigned through the use of the DI card in the control section of the `*.pre` file (see section 10.2.25).

Apart from these general formulations, there are a number of special solution methods for dielectric bodies in FEKO:

- Thin dielectric sheets:
  The volume equivalence principle is applied and the resulting equivalent currents approximated by a surface current (see the SK card, section 10.2.42).
- Dielectric coatings:
  Metallic wires or triangular surface patches can have some dielectric coating (see the CO card, section 10.2.23).
- Dielectric half-space e.g. ground surface:
  In this case the reflection coefficient method is used (see the BO card, section 10.2.19).
- Spheres consisting of one or more dielectric layers:
  A special Green's function is available (see the GF card, section 10.2.30).
- Planar multilayer substrate (with or without a perfect conducting ground plane):
  A special Green's function is available (see the GF card, section 10.2.30).

The entered structures e.g. metallic wires and surfaces, do not necessarily have to be embedded in free space. The EG card (section 9.2.12) or DI card (section 10.2.25) or also the GF card (section 10.2.30) can be used to specify the material parameters of the surrounding medium.

2.6 Dynamic memory management

2.6.1 Telling FEKO how much memory can be used

FEKO has the ability to manage the memory dynamically, i.e. the memory required for the geometry data and matrix elements etc. is determined and allocated at run time. When FEKO tries to allocate memory, in principle the operating system offers a certain address space, which might either be physically installed memory (i.e. RAM), but also virtual memory (system swap space swapped to the hard disk).

If FEKO starts to swap using virtual memory, then the whole solution process can be slowed down quite significantly, and this is not recommended. FEKO also has an out-of-core solution which uses the data on disk in a much more efficient way. (The out-of-core technique is also used, of course, if the problem requires more memory than is available in both RAM and virtual memory.)

For solutions which do not fit into the available RAM, but do fit into the RAM plus virtual memory, the user should tell FEKO what the actual physically installed memory is, less a certain reserve for the operating system.

Up to FEKO Suite 4.2 the variables `#maxalloc` or `#maxallocm` were used for this purpose:

`#maxallocm`	This sets the maximum allocatable memory in MBytes. For example the definition `#maxallocm = 400` will allow a maximum of 400 MByte of memory to be allocated. If this is not enough, the matrix will be saved to the hard disk or the program will be halted. For parallel versions of FEKO this memory limit applies to each process. Note that if `#maxallocm` is used, it will have preference, and `#maxalloc` will be ignored if it is used in the same `*.pre` file.
`#maxalloc`	Similar to `#maxalloc` but just specifies the memory in Bytes and not in MBytes.

These two variables `#maxalloc` and `#maxallocm` are still supported in FEKO Suite 4.2 and later for backwards compatibility reasons, however, their usage is strongly discouraged now. Rather on Windows and Linux the FEKO memory management works fully automatic and the usage of virtual memory (swap space) is avoided. On other UNIX versions (such as HP-UX or SUN Solaris) during the installation an environment variable `FEKO_MAXALLOCM` is defined and gets set in the `initfeko` script, which specifies the physically installed RAM less an operating system reserve for a specific machine. The

advantage then is that the FEKO `*.pre` files are machine independent, i.e. if you have two different computers with different memory, no variable `#maxallocm` has to be changed in the `*.pre` file as such, everything is either fully automatic (Windows or Linux) or defined on a per machine basis (other UNIX versions.)

2.6.2 Other variables that are under user control

In some cases memory blocks have to be allocated for data storage before FEKO knows exactly how big these memory blocks have to be. In these cases it uses an estimate calculated by PREFEKO. If the estimated size is too small, FEKO will stop execution and give an error message. The appropriate size now has to be declared in the `*.pre` file. This is done by entering, for example, the line `#maxnv = 100` as for normal variables — see also section 7.3 — anywhere in the `*.pre` file.

`#maxnp`	The maximum number of columns and rows which a block in the matrix consists of in the Block-Gauss algorithm which solves the matrix equation. Dynamically `3*#maxnp*#maxnp*16` Bytes are allocated for 3 blocks in the matrix.
`#maxnv`	The maximum number of connection points between wires and surface triangles.

2.6.3 Variables that are automatically set correctly

Note that normally PREFEKO estimates the following variables correctly and they should only be declared in cases where there is an explicit error message stating that larger memory blocks are required. Under normal circumstances these variables should not be set, they could have a negative impact on the FEKO performance.

`#maxaeedges`	The maximum number of edges between triangles that may be excited with the AE card.
`#maxanr`	The maximum number of sources.
`#maxapo`	The size of the memory block that is used to save the coefficients in the physical optics approximation. For `#maxapo=0` the necessary amount will be dynamically allocated.
`#maxarang`	The maximum number of ϑ or φ angles used with the AR card (excitation by a point source with a specified radiation pattern).
`#maxarpat`	The maximum number of radiation pattern excitations (AR card) allowed simultaneously.
`#maxbsobnr`	To accelerate the ray path search with PO the area under consideration is divided into boxes. Information pertaining to which box contains which object must be stored. A field of size `#maxbsobnr` is used in this case.

`#maxcolayer`	The maximum number of layers on a CO card which implements thin dielectric sheets.
`#maxdrnv`	The maximum number of triangle elements that can be connected to a segment at an attachment point.
`#maxfepkts`	The maximum number of points considered for the near field computation with the FE card when using *Specified points*.
`#maxfoge`	The maximum number of areas that are described by using the Fock theory.
`#maxgfmsia`	The maximum number of entries in the interpolation tables for the Green's function of a planar substrate.
`#maxhacards`	The maximum number of HA cards (used internally to set up microstrip ports) that may be present in the `*.fek` file.
`#maxkanr`	The maximum number of "internal" edges (also the number of basis functions) per triangle. It may be larger than 3 if more than two plates share an edge.
`#maxknonr`	The maximum number of nodes that may lie against a segment.
`#maxl4cards`	The maximum number of L4 type loads.
`#maxlab`	The maximum number of labels in a model.
`#maxlecards`	The maximum number of LE cards (which specify a load on an edge between triangles).
`#maxleedges`	The maximum number of edges between two surface triangles that can be loaded with a single LE card.
`#maxlengz`	Dimension of the interpolation table used for the planar multilayer Green's functions. This variable determines the maximum number of sample points in the z direction.
`#maxmedia`	The maximum number of different media used for the treatment of dielectric bodies in the surface equivalence principle. The surrounding free space (medium 0) is not counted (i.e. with `#maxmedia=1` one dielectric body can be treated).
`#maxndr`	The maximum number of triangles.
`#maxnka`	The maximum number of edges between two triangles.
`#maxnkapo`	The maximum number of edges in the physical optics approximation.
`#maxnkno`	The maximum number of nodes between segments.
`#maxnlayer`	The maximum number of layers for the special Green's function of a planar substrate.
`#maxnqua`	The maximum number of dielectric cuboids.
`#maxnseg`	The maximum number of segments.
`#maxntetra`	The maximum number of tetrahedral volume elements for a FEM solution.
`#maxnzeile`	The maximum number of basis functions in the moment method area.
`#maxpoka`	The maximum number of bordering edges to the PO area.

#maxpokl	The maximum number of wedges in the PO area.
#maxpolyf	The maximum number of polygonal surfaces that can be used to represent a body in the UTD region.
#maxpolyp	The maximum number of corner points allowed for a polygonal plate.
#maxpovs	The maximum number of label to label visibility specifications set by VS cards (a card with a range sets a number of entries equal to the size of the range).
#maxsklayer	The maximum number of layers at an SK card.
#maxtlcards	The maximum number of TL cards.
#maxutdzyl	The maximum number of cylinders in the UTD region.
#nmat	The memory size that may be allocated for the matrix of the system of linear equations. For #nmat=0, the necessary amount will be allocated dynamically. The allocation is not specified in Bytes, but in terms of the number of type DOUBLE COMPLEX numbers. (These require 16 Bytes each.) For example, 400 MByte is specified by setting #nmat = 400*1024*1024/16. The same effect can be achieved by setting the variable #maxalloc such that it is unusual to assign a value to #nmat.
#npuf	The maximum number of control cards that may occur in a loop (for example in a frequency sweep).

2.7 Summary of the files

The table below gives an overview of the different files and their respective functions. STDOUT is the standard output, usually the screen. A * is used to symbolically indicate the filename.

*.pre

Filename	Description
STDOUT	Usually the screen. This is where comments such as progress, warnings and errors are sent.
*._14	Page (temporary storage) file for the matrix in the sequential version of FEKO with out-of-core solution.
*._15	Page (temporary storage) file for the matrix in the sequential version of FEKO with out-of-core solution.
*._16	Page (temporary storage) file for the matrix in the sequential version of FEKO with out-of-core solution.
*._20	Page (temporary storage) file for the coupling coefficients of the MoM/PO hybrid method during an out-of-core solution.
*._21	Page (temporary storage) file for the coupling coefficients of the MoM/PO hybrid method during an out-of-core solution.
*.afo	Continuous frequency results created by ADAPTFEKO.

`*.aus`	Output file of TIMEFEKO.
`*.bof`	Binary version of the output file which is used for post processing.
`*.cdb`	ANSYS mesh file which can be imported with the IN card.
`*.cfm`	CADFEKO model file (exported CADFEKO mesh to be imported by PREFEKO).
`*.cfs`	Session file of CADFEKO.
`*.cfx`	Native CADFEKO geometry file (contains geometry and mesh etc.)
`*.cgm`	Contains the size of the residue that results from the iterative algorithm which solves the matrix equation and the number of iterations. This file is only generated on request by a DA card (section 10.2.24).
`*.dbg`	When using the UTD, it is possible to request an optional output file containing a large amount of additional data (and may therefore be very large), see the UT card.
`*.dxf`	AutoCAD geometry file which can be imported with the IN card. (PREFEKO can import arbitrary surfaces from meshed `*.dxf` files. It can also import and mesh lines and polyline surfaces, see section 9.2.18.)
`*.edg`	Geometric data is taken from the `*.fek` input file, where for example common edges between triangles are found. This reprocessed information of the geometry is saved in the `*.edg` file. (EG card, section 9.2.12)
`*.efe`	File containing the electric field strengths. Contains both the position and the complex components of the electric field strength vectors. This file is only generated on request by a DA card (section 10.2.24).
`*.fek`	Output file from PREFEKO — serves as the input file for FEKO.
`*.ffe`	File containing the far field data. This file is only generated on request by a DA card (section 10.2.24).
`*.gfe`	Interpolation table of the electric field strengths for the Green's function of a layered sphere.
`*.gfh`	Interpolation table of the magnetic field strengths for the Green's function of a layered sphere.
`*.hfe`	File containing the magnetic field strengths. Contains both the position and the complex components of the magnetic field strength vectors. This file is only generated on request by a DA card (section 10.2.24).
`*.inc`	Include file for PREFEKO.
`*.isd`	Data file containing the field distribution calculated by FEKO for coupling with CableMod or PCBMod.
`*.log`	Log file created by OPTFEKO.
`*.lud`	File in which the elements of the LU-decomposed MoM matrix are stored (only generated on request of a PS card, section 10.2.39).
`*.mat`	File in which the matrix elements of the linear equation system, are stored (only generated on request of a PS card, see section 10.2.39).
`*.nas`	NASTRAN geometry file which can be imported with the IN card.
`*.neu`	Geometrical data file which is exported by the program FEMAP.
`*.opt`	Input file for the program OPTFEKO.

`*.ofc`	Paging files for the array elements used with sequential and parallel out-of-core solution. (To avoid the 2 GByte file size limit; or on parallel systems with a distributed file system, several files may be used. These are distinguished by adding numbers to the filename.)
`*.os`	File containing the surface currents and the currents in the segments. Data is in the form of position and the complex components of the current density vectors. This file is only generated on request by a DA card (section 10.2.24).
`*.out`	Output file from FEKO in which the results of all the calculations and messages can be found.
`*.pcr`	Exported LU preconditioner for the FEM.
`*.pfg`	POSTFEKO graph file.
`*.pfs`	POSTFEKO session file.
`*.pre`	Input file for PREFEKO.
`*.ray`	For the UTD an optional ray file can be requested. This file is not required when visualising rays in POSTFEKO.
`*.rhs`	File containing the right hand side vector in the system of linear equations.
`*.rsd`	File for coupling of FEKO with CableMod or PCBMod. It is usually created by CableMod or PCBMod, but can also be created by FEKO if requested with the OS card (field calculation along lines).
`*.sha`	File storing shadowing information for the PO.
`*.snp`	Touchstone format S-parameter file as created by the DA card. The `n` refers to the number of ports (see section 10.2.24).
`*.sph`	Spherical wave expansion (SWE) as used by the reflector antenna code GRASP from TICRA (can be exported in FEKO using the DA card, see section 10.2.24).
`*.str`	File in which the coefficients of the basis functions) are stored for reuse (generated on request from a PS card, section 10.2.39).
`*.tim`	Input file for the program TIMEFEKO.
`*.vis`	When multiple reflections are used with the PO formulation FEKO determines which basis functions have line of sight visibility. Since this calculation may require significant run time, this information can be saved to a `*.vis` file for reuse.
`*.wfg`	Figure created and saved with GraphFEKO.
`*.x_t`	Parasolid model file.

The files `*.efe`, `*.hfe`, `*.ffe` and `*.os` are redundant. All the information in these files is also available in the `*.out` file. The format of these redundant files lends itself more readily to further processing.

2.8 Environment variables

This section lists the environment variables that may be used to control the execution of FEKO. See also the discussion of the installation (the script `initfeko` or the batch file `initfeko.bat`) in the *Getting started manual*. This script and batch file is usually automatically created by the FEKO installation program, and the environment variables are set correctly. Therefore the user does not need to set environment variables manually.

The following environment variables may be set.

`FEKO`	Must be set to the path where the FEKO executables are located, which normally is `FEKO_HOME/bin`.
`FEKO_CMDINFO`	If this environment variable is set to the value 1, FEKO writes additional data concerning the number and the value of the received command line parameters to the screen. This can be useful to trace errors in the parallel version of FEKO used in connection with some implementations of `mpirun`, `mpiopt`, `mpprun` etc.
`FEKO_HOME`	This variable is set to the FEKO installation path which contains the subdirectories such as `bin`, `doc`, `license` and, for the parallel version, `mpi`.
`FEKO_LITE`	If this environment variable is set to 1, then FEKO runs for 30 days in the un-registered FEKO LITE mode. FEKO is then restricted regarding the element sizes etc., but does not need a licence (see the *Getting started manual* for more details).
`FEKO_MACHFILE`	The parallel version of FEKO is started by running RUNFEKO with options `-np x`. When FEKO is installed on a parallel computer or a computer cluster, the configuration of the cluster and the number of processes that should be run on each computer is specified during the installation. This can be overwritten for any FEKO run by creating a so-called *machines file* and setting the environment variable `FEKO_MACHFILE` to point to this file. (More detail can be found in section 8.2.)
`FEKO_MACHINFO`	If this parameter is set, FEKO will write information about the machine precision to both the screen and the output file.

FEKO_MAXALLOCM	This environment variable is used to limit memory (in MByte) that FEKO is allowed to use on this computer. This environment variable is not needed or recommended for computers running Windows or Linux operating systems. On others the variable is set at installation time. The value of this variable should usually be set equal to the physical memory minus 70 MBytes for the operating system. In a few cases a lower limit may be required, and should be set here.
FEKO_MPISTATISTICS	This environment variable provides additional information about the performance of the parallel version of FEKO. There are three options 1 Give a detailed report of the CPU and run times for the individual processes. It is, for example, possible to determine how much time each process required during the computation of the array elements. 2 Give as additional output the MFLOPS rate of each process (without network communication time). This is useful to determine the relative performance of nodes in a heterogeneous cluster. 4 Give information about the network performance (latency and bandwidth). This is very useful when configuring parallel clusters. The options can be added in a binary fashion, for example setting FEKO_MPISTATISTICS=5 will print both the run times and network performance.
FEKO_PARALLEL_DEBUG	For parallel runs of FEKO under UNIX, this environment variable can be set to 1 in order to see all the details and commands used in the parallel launching and machines file parsing etc. This is helpful for troubleshooting errors.

FEKO_RSH	When installing the parallel FEKO version on a UNIX cluster, then communication between the nodes is required both at installation time (checks on the remote nodes, remote copying of files, remote execution of utilities etc.), but also when using FEKO (remote launching of parallel FEKO processes). By default both the installation script and the parallel launcher will use the remote shell for this purpose (`remsh` under HP-UX and `rsh` for all other UNIX platforms). A typical setup is then to use a `~/.rhosts` file (see detailed comments given during the installation). But this is not quite secure, and one might prefer to rather use the secure shell `ssh` in connection with public key authentication (avoids having to type passwords all the time). The actual remote shell executable (e.g. `rsh` or `remsh` or `ssh`) will be determined during the installation procedure, and the environment variable `FEKO_RSH` will be set to point to this executable. This can always be changed later (e.g. using `rsh` for the installation as root, but then `ssh` for the users using the parallel FEKO version or vice versa) by changing the value of the environment variable `FEKO_RSH` in the FEKO initialisation file `initfeko` or `initfeko.csh` accordingly. One can also set this on a user per user basis, then directly for instance in `~/.profile` after having executed `initfeko`. This environment variable will not be used on Windows systems.

FEKO_TMPDIR	This variable specifies the directory where FEKO will write paging files, when using the out-of-core solution. In the past it was required that the definition ended in a backslash (Windows) or a slash (UNIX). This is no longer required. For example, in UNIX it may be set with
	`set FEKO_TMPDIR=/tmp`
	`export FEKO_TMPDIR`

FEKO_USER_HOME	This directory is used to write user specific initialisation files. This variable replaced `FEKO_WRITE`. It is provided to allow different users to save unique configurations, or for when the user does not have write access to the `FEKO` directory. For Windows systems this is normally the same as `FEKO` and on UNIX systems it is usually set to `$HOME\.feko` during the installation.

FEKO_WHICH_MPI

FEKO uses different MPI implementations for the different platforms and thus the different platforms require different command syntax to start FEKO. RUNFEKO provides an interface that remains the same on all platforms. However, it must know which MPI implementation is used. This is done by setting the environment variable FEKO_WHICH_MPI (it is automatically set by the installation script) to one of the following options:

0	Not initialised
1	MPICH (for example Linux or SUN)
2	HP-MPI
3	NEC-MPI
4	SGI-MPI
5	CRAY/SGI-MPI on CRAY T3E
6	ScaMPI on Linux IA32, AMD64, EM64T
7	Compaq MPI on Tru64 UNIX
8	SCore on Linux Myrinet clusters
9	GM on Linux Myrinet clusters
10	Parastation MPI on Linux IA32

FEKO_WRITE_RHS

If this environment variable is set (value arbitrary), FEKO writes the right side of the set of linear equations to a *.rhs file. This is only useful for test purposes, such as when one wants to analyse this vector with another program.

FEKOLANG

Selects the language of operation. This must be either d for German, or e for English.

MKL_SERIAL

If this is set to YES, FEKO (and all other codes using the Intel MKL libraries) will run as a single threaded application (i.e. it will not utilise multiple CPUs) irrespective of the value of OMP_NUM_THREADS (see below).

OMP_NUM_THREADS

Sequential FEKO versions for Windows or Linux PCs (on Intel or compatible) can use multiple CPUs on one board for the LU-decomposition of the MoM matrix (the parallel FEKO version will have all phases of the solution in parallel). To use this, the environment variable OMP_NUM_THREADS must be set to the number of CPUs to use. (See also MKL_SERIAL above.)

2.9 Checking the validity of the results

If a calculation has been done with FEKO, the results have to be checked. There are a number of ways of doing this:

- a comparison with exact results, if these are available
- a comparison with results that have been published in the literature
- a comparison with another program that is based on another method of calculation.
- a comparison with measured results
- plausibility, e.g. negative real input impedances do not exist.

If these possibilities are not available, then the following should be tried:

- After a normal calculation with FEKO, repeat it with a finer mesh (see IP card, section 9.2.19). The number of segments should be at least 1.5 times greater than with the normal calculation. If there is a large difference in the results, then an error has occurred.
- Doing a power balance. The power fed into an antenna through the power source must be equal to the radiated power. The radiated power can be calculated by integrating the power flux density. This is done by using the FF command (see `example_07` or `example_08` in the *Examples Guide*).

3 The licence manager

The FEKO licence manager shows all the licences in the specified `secfeko.dat` file (for nodelocked licences) or connects to the floating licence server and retrieves licence information from there. This allows viewing the active components / modules, limits and expiry dates of all licences in a clear human readable form. In addition — for floating licences — it allows viewing the available licences, selecting which licence to use, and managing the licences that are in use. See also the section on floating licences in the *Getting started manual*, in particular if you still need to set up the floating licence server.

The licence manager is started by typing `secfeko_gui` from a console window. Under MS Windows it may also be started by selecting *Licence manager* from the *FEKO* menu under the Windows *Start* menu.

3.1 Displaying licence information

Figure 3-1 shows the licence manager with a typical node locked licence. If the splitter between the floating and node locked licences is dragged upwards, the area used for floating licences can be reduced or even removed completely. Drag it downward to show the floating licence area again.

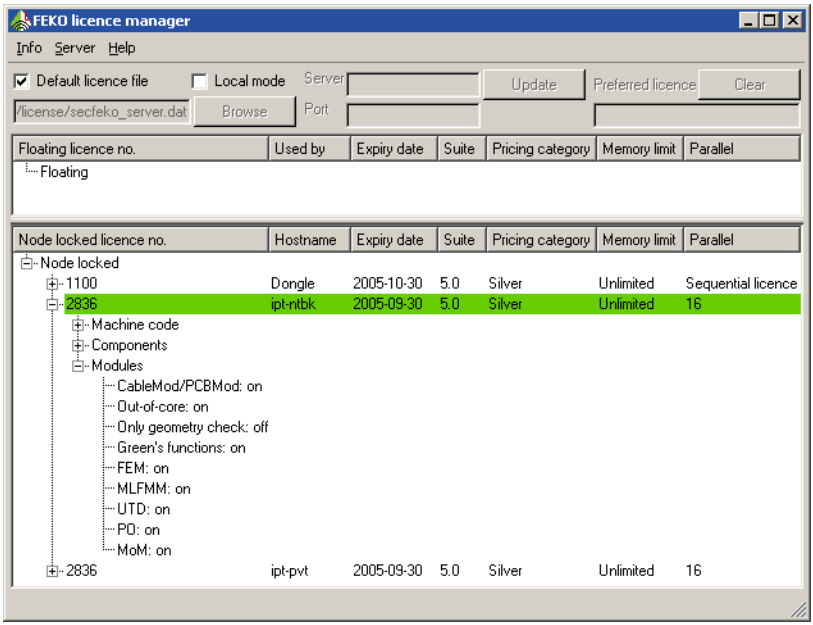


Figure 3-1: The licence manager with a typical node locked licence

The licence manager will, by default, open the file `FEKO_HOME\license\secfeko.bat` which is also used by all FEKO components. The user may, however, also uncheck *Default licence file* and browse for an alternative file, for example to check the information in a new `secfeko.dat` file before replacing an existing licence file. If *Local mode* is checked, floating licences are displayed without connecting to the server.

The licence for the current host is shown in green. Note that FEKO will always use the first valid licence, i.e. there should generally only be one valid licence on each machine. Expired licences are shown in red.

The main table displays — for each licence — the hostname (if available) where the licence is valid, the expiry date, the highest allowed version (one can usually use older versions with newer licence files, but not the other way around), the pricing category, the memory limit if there is one, and the number of parallel processes allowed if applicable.

Each licence also contains the following information (click on the icon to the left of the licence to expand / contract it):

- *Machine code:* Expanding this item shows the machine code of the machine where this licence is valid. (Only shown for node locked licences.)
- *Components:* This item lists all the components (for example, EDITFEKO, FEKO, RUNFEKO, etc.) which are activated in this licence.
- *Modules:* The modules and extensions that are active, for example if you purchased the CableMod/PCBMod interface there will be an entry `CableMod/PCBMod: on`.

3.2 Managing floating licences

Figure 3-2 shows the licence manager with a typical floating licence. If the `secfeko.dat` file contains a floating licence server, the server and port number are displayed at the top of the licence manager. The *Update* button allows the user to update the state of the licences. (Note that floating licences are checked in and out as they are being used, thus this information is not static as is the case with node locked licences.)

With the exception of the machine code of each licence, all the information displayed for node locked licences is also applicable for floating licences. Here, however, a green line indicates a preferred licence (see section 3.2.1 below) and a light blue line indicates a licence that is in use.

For each licence that is in use the table displays the user and hostname where this licence is used and the *Parallel* column indicates how many parallel processes are available. (If a user has a 16 node licence, he may start two 8 process parallel jobs, but not an 8 process job and a 10 process job.) In addition, expanding the licence shows information on when the licence was checked out and what components are in use. If one checks in a licence (see section 3.2.2 below) which has just EDITFEKO checked out the user may be able to still save his work, but if there is an instance of FEKO checked out it will be terminated.

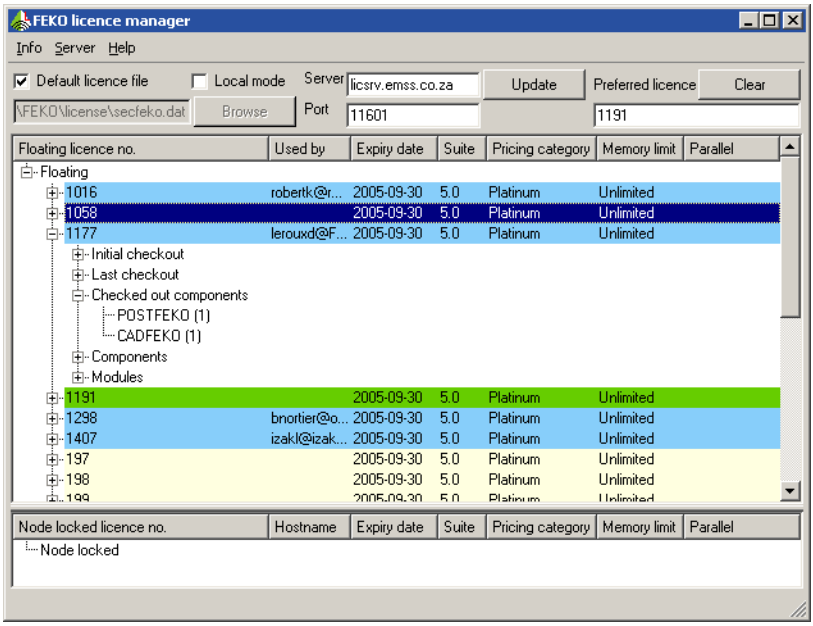


Figure 3-2: The licence manager with a typical floating licence

3.2.1 Preferred licence

When a new licence is requested, the floating licence server will check out the first available licence. Where users have multiple licences they may not necessarily be identical. (Some licences may, for example, be memory limited GUI licences, or the CableMod/PCBMod module may be activated only in some licences.) In this case the first available licence may not be the desired one.

The licence server thus supports a preferred licence. Right click on any licence and select *Mark as preferred* from the pop-up menu. The preferred licence is listed at the top right of the licence manager. When the user next requests a new licence, this one will be used. (The preferred licence can also be selected by clicking on the licence and selecting *Server → Set preferred licence* from the main menu.)

Click the *Clear* button or select *Server → Clear preferred licence* from the main menu to remove the preferred licence. Since the server will just use the first available licence if the preferred licence is already in use, there is very little need for an individual user to clear his preferred licence. However, if your company has one full and a number of limited GUI licences you should not check out the full licence while just setting up your model — then nobody else can use this and you could possibly have used one of the GUI licences.

Finally, note that all licence requests from the same user and the same machine will

use the same licence. Thus if you set a preferred licence while you have other FEKO components running (the licence manager does not require a licence), the new licence will only be used once you have closed all components and opened a new one.

3.2.2 Managing the server

While a licence is checked out, nobody else can use that licence. It may happen that a user has say CADFEKO open on his machine while he is on leave, or that the licence is urgently needed for another run and the user is not available to stop his run. The licence manager therefore allows checking in a licence such that it becomes available again. Right click on the licence and select *Check in licence* from the resulting pop-up menu, or click on the licence and select *Server* → *Check in licence* from the main menu. For security reasons, this check in operation is allowed only for users with administrative priviledges (Windows) or the root user (UNIX) or the user who has initially checked out this particular licence.

If a licence is checked in, all components using that licence will fail the next time they check the licence server. In the case of the solver components (RUNFEKO, PREFEKO, FEKO, ADAPTFEKO, OPTFEKO and TIMEFEKO) the error is fatal and the entire solution will be lost. In the case of EDITFEKO, the user will get an option to switch to the FEKO LITE mode and can still save his work. Thus it is usually preferred to check in a licence where only EDITFEKO is checked out.

In addition, the *Server* item on the main menu allows *Shutdown* and *Reset*. For security reasons, both these operations are restricted to users with administrative priviledges (Windows) or the root user (UNIX) or the user account under which the licence server is running. Generally it should only be necessary to *Reset* the licence server when modifying the licence file (for example when obtaining a new licence file from EMSS or when changing the port). *Note that resetting the licence server will check in all licences,* such that all active user jobs will be terminated similar to when checking in individual licences. Users should not need the *Shutdown* option. This is only required to stop the service such that the files can be deleted, but this will usually be done by the installation program. Note also that once the licence server has been shut down it must be restarted on the server itself — the licence manager cannot connect to the service if it is not running. Please check the *Getting started manual*, for further instructions on the floating licence server installation and startup.

3.3 Determine the machine code

Finally, the machine codes of the current machine can be displayed by selecting *Info* → *Machine code info* from the licence manager main menu. All licences that match any of these machine codes will be valid on this machine.

4 Working in CADFEKO

4.1 Introduction

CADFEKO has been developed to allow the creation and set up of FEKO models in a graphical or CAD environment. This involves defining and meshing the geometry of the problem as well as specifying the electromagnetic parameters and solution requirements. Currently CADFEKO is used for the geometry construction and meshing phase and EDITFEKO is used to define the solution requirements. (See the *Getting started manual* for more details.)

CADFEKO supports parametric model construction. If the model is constructed using variables, the entire model can be modified by changing these variables. This is used, for example, to tune the size of an antenna for a required frequency and input impedance. In addition, CADFEKO maintains the construction history. This allows, for example, the modification of individual objects used in a union operation after the operation. CADFEKO can also import and mesh complex CAD models in Parasolid, ACIS, Catia, ProEngineer, STEP and Unigraphics formats as allowed in the licence. It is important that the imported model is of a decent quality. If the imported model has overlapping surfaces and gaps between supposedly connected surfaces, so will the resulting CADFEKO model. Refining such models is a considerable task.

CADFEKO is available on MS Windows 2000/2003/XP and on Linux systems using glibc 2.3 or later.

4.1.1 Starting CADFEKO

If CADFEKO is properly installed as part of the FEKO installation, it can be started from the FEKO folder in the MS Windows *Start* menu, or by opening a console and typing `cadfeko`. It is possible to specify a filename on the command line, for example, `cadfeko model`. Only CFX files (i.e. CADFEKO models) or their associated CFS (workspace) files are allowed here. The filename may be specified with or without extension. If the file does not exist, CADFEKO creates a new model provided the current / specified directory does not contain a CFS, PRE or CFM file with this name. (If such files exists, CADFEKO will start with a blank model.) On MS Windows any CFX file can be opened by double clicking on it.

4.1.2 Display settings and graphics cards

CADFEKO can be used with 256 colours, but a colour setting of at least 16 bits is recommended. In addition, practical use of CADFEKO will require a screen resolution of 1024×768 or larger.

CADFEKO uses OpenGL for rendering and this depends on the graphics card being used and the drivers installed for it. With a properly installed graphics card that supports 3D hardware rendering, a considerable speedup in visualisation should be seen. (See the graphics card's documentation for information on OpenGL support.) Note that some cards only support OpenGL at a certain colour depth.

While CADFEKO may work without trouble on a large number of graphics cards, the following cards have been tested and are recommended:

MS Windows	
ATI	Radeon 800X Pro
	Radeon 9000
	Radeon 9800 Pro
	Radeon 9600 XT
	Radeon 8500
	Radeon X600 XT PCIe
	Fire GL Z1 (VPU FGL9500)
	Fire GL V3100 PCIe
NVidia	Quadro 4 900XGL
	Quadro 4 980XGL (PNY)
	Quadro 4 580XGL (PNY)
	Quadro FX 1100 (PNY)
	Quadro FX 3000 (PNY)
	Quadro FX 1300 (PNY) PCIe
	GeForce PCX 5750 (Jaton) PCIe
	GeForce 6600GT (Jaton) PCIe
	GeForce 6800 Ultra (Gigabyte) PCIe
	GeForce FX 5200
	GeForce FX 5700 (Gigabyte)
	GeForce FX 5950 Ultra (PNY)
3Dlabs	Oxygen GVX1
Linux	
ATI	Radeon 800X Pro
	Radeon 9800 Pro
	Radeon 9600 XT

In addition, CADFEKO can be configured to allow accurate rendering on most graphics cards. Select *Options* → *Rendering* from the main CADFEKO menu to open the *Rendering options* dialog (see section 4.3). Some cards may give better (but slower) results using software rendering. Face displacement allows a trade-off between edges appearing broken and supposedly hidden lines being visible.

4.2 CADFEKO overview

The CADFEKO display, shown in figure 4-1, contains — in the default configuration — a main menu and toolbar (split into the *Standard* and *Selection* toolbars) at the top, a toolbar (split into the *Create geometry* and *Modify geometry* toolbars) and tree view on the left, a message window and status bar at the bottom, and a window area (containing the 3D views and/or the notes editor).

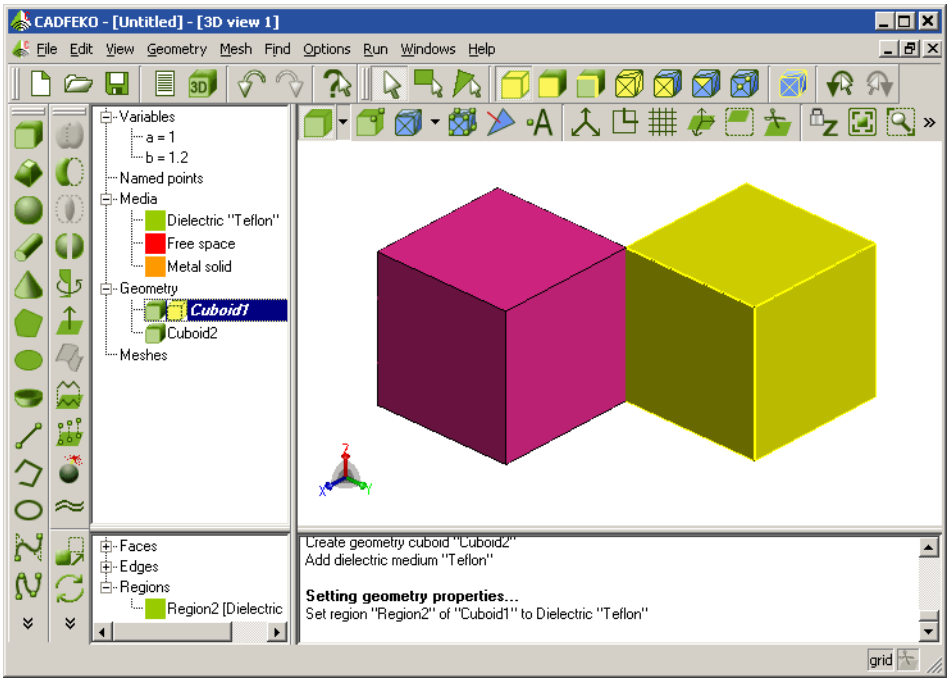


Figure 4-1: The main CADFEKO display

The message window displays messages about user interaction such as geometry creation and undo / redo commands. It also shows more details about error messages. The status bar shows the snap mode and the coordinates of the current cursor position — in global coordinates — using the selected snap mode. This works with standard point entry (see section 4.6.3), or by pressing <Ctrl><Shift> while moving the mouse in the 3D view.

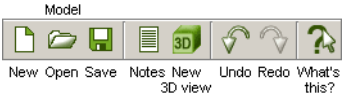
The tree view and message window can be resized by dragging the splitter bars. The panels can be collapsed by dragging the splitter all the way to the edge and shown again by dragging the splitter away from the edge. When the tree (or one of its halves) or the message window is collapsed, the *View* menu contains *Tree* or *Messages* which can be used to show them again. The toolbars can be docked at any of the four sides or dragged free into their own windows (click and drag the double lines on the top or left). When a

toolbar is dragged to a location where it may be docked, it is drawn with a thin outline instead of the normal border. The toolbars can be shown / hidden from the *View* menu or by right clicking on the main menu and selecting the appropriate item.

CADFEKO strives to be as unrestrictive as possible. For example, the workplane and the view orientation may be changed, or a new variable defined, while creating geometry without having to close the geometry creation dialog first. If a new dialog is opened while another dialog is open, all the input fields on the first dialog are disabled until the new dialog is closed. If a number of dialogs have been opened, they will become available in reverse order as each dialog is closed. The dialogs themselves are not disabled, thus they can still be moved on the screen. (Some operations, such as modifying a variable, may change the model. These operations are not allowed while working with other tools that operate on the model. In addition, a second dialog cannot be opened for the same item. For example, if a geometry creation dialog is opened while the *Edit workplane* dialog of a 3D view is open, the workplane for that view cannot be edited without closing the geometry dialog and going back to the open workplane dialog.) If the items a dialog applies to are no longer selected, they are shown in green in the tree.

Generally items are selected before applying operations to them.

4.2.1 The standard toolbar



This toolbar contains short-cut options to the file menu items used to create, open and save models. The *Notes editor* item opens the notes view (see section 4.17). The *New 3D view* item (also available under the *View* menu and with the short-cut key <Ctrl><3>) creates a new 3D display window. (All views apply to the same model.)

The *Undo* and *Redo* items (these are also available under the *Edit* menu and the short-cut keys <Ctrl><Z> and <Ctrl><Y>) apply to actions on the model, i.e. all items that affect the model. (Typically these include creating and modifying geometry, adding / modifying / deleting variables, changing labels, meshing, etc.) If a variable is used in a workplane, the model undo list is blocked after the creation of the variable. To undo beyond that point, the variable must first be removed from the workplane. View operations (such as rotation, zooming, etc.) have a separate undo / redo list for each view. There is also a separate list for undo / redo of the selection (shared by all views). The undo lists are not saved with the model, i.e. if the model is saved and loaded, the lists will be cleared.

If an object is remeshed or a mesh is deleted, the entire existing mesh is stored in the undo / redo list. This allows quick undo / redo, but can require significant memory if

large meshes are modified a few times. Therefore it is recommended that the number of undo steps be reduced during extensive mesh manipulation. (See section 4.14.)

To obtain additional information on any component of the user interface, click the *What's this?* button and then click on the component. This provides some help on a level between the very brief tool tip and the full online manual.

4.2.2 Creating models

Select *File* → *New* to remove the entire geometry, the mesh and any notes from the model and start a new model with a clean environment. Similarly, select *File* → *Open* or *File* → *Save* to open and save CADFEKO models. Only the CFX file is required to open the model, but when saving the model, CADFEKO automatically saves the CFS file — containing the workspace (views, workplanes, etc.) — and the CFM and PRE files — used when solving models (see section 4.15). The *File* → *Save as* option saves the current model (CFX, CFS, CFM and PRE files) to a new name. In addition to saving the files, this option updates the PRE file to refer to the new CFM file.

File → *Export* → *CFM file* exports the current mesh to a CFM file which can then be imported in a PRE file. The mesh is exported as is, i.e. there are no checks to see if it is valid at this point. If there are no mesh elements, variables or named points in the model, the CFM file will effectively be empty and importing it in a PRE file will have no effect. This menu item only exports the mesh and neither the name nor the file itself is used again by CADFEKO. When running components from the *Run* menu, CADFEKO will automatically export a CFM file using the model name.

4.3 The 3D view

The 3D views are used to display and interact with the model. CADFEKO distinguishes between a click operation (mouse down and up at approximately the same location) and click and drag (the mouse is moved a minimum distance while the button remains down). Clicking is used for selection (see section 4.11), opening a context-sensitive menu with operations for the current selection (right mouse button) and point entry (left mouse button, see section 4.6.3).

The model is rotated by clicking and dragging the mouse. It can also be zoomed by pressing <Shift> while clicking and dragging the mouse (upward movement zooms in, while downward movement zooms out; horizontal movement is ignored). Rolling the mouse wheel also zooms. (Pressing <Shift> while rolling the mouse wheel slows down zooming.) CADFEKO tries to zoom in / out on the object below the mouse pointer, hence rolling the mouse wheel while the cursor is near the edge of the screen will also have a panning effect. After a zoom operation the model will rotate around a point that is more or less in the middle of the screen at the same distance as the object zoomed on.

Finally the display can be panned by clicking and dragging while pressing the <Ctrl> key or clicking and dragging with the middle mouse button.

The direction of the face normal is often important. For example, an edge between two surfaces with opposite normals is not redundant and cannot be deleted. For this reason the front (normal) and back sides are coloured differently in the default colour view (see section 4.3.4) and it is possible to reverse face normals (see section 4.7.8).

It is possible to hide (see section 4.3.6) specific items. If an item is hidden, it is removed from all 3D views independent of the setting on the 3D view toolbar (see section 4.3.4).

In some cases the display may show items that should be hidden. These rendering problems are very much hardware dependent and CADFEKO allows some configuration in this regard. Select *Options* → *Rendering* to open the dialog shown in figure 4-2. The *Rendering mode* group shows the algorithm used to remove hidden lines. If *Auto-detect* is unchecked, it can be configured manually. This may improve the display accuracy, but it can have a significant impact on the rendering performance and memory usage. *Hardware rendering* is only enabled when available on the user's system. The sliders change the face displacement for geometry and meshes. Setting the face displacement allows a trade-off between edges appearing broken and supposedly hidden lines being visible. The relative positions of the sliders also determine what is visible if both the geometry and mesh are displayed. Different settings may be required for different view directions. These options are stored in the CADFEKO configuration file.

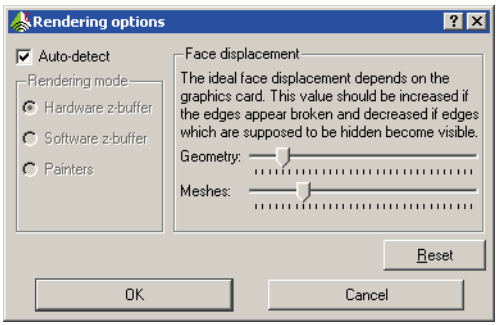


Figure 4-2: *Rendering options* dialog

4.3.1 The workplane

Each 3D view has an associated workplane which is used to specify 3D coordinates with the mouse. The workplane is drawn as a blue rectangle (on the boundary) and the U, V and N axes. The workplane display can be switched off by clicking *Show / hide workplane boundary* on the 3D view toolbar. This applies to the current view only and will also hide

the grid. (There is also a separate button to hide only the grid.) To set the position / orientation of the workplane, right click on the 3D view and select *Workplane → Edit* or press <F9> while the view has focus. (There is also an *Edit workplane* button on the 3D view toolbar.) This opens the dialog in figure 4-3.

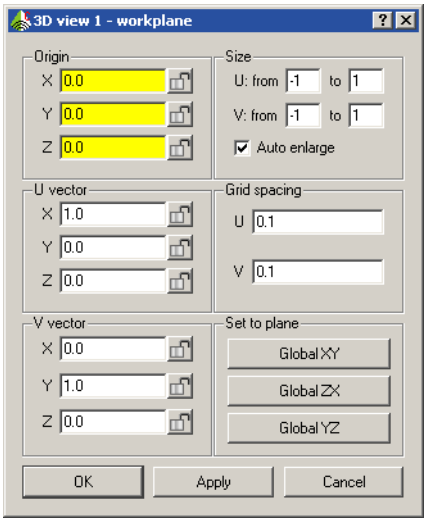


Figure 4-3: The workplane dialog

The workplane is specified in terms of an *Origin* which defines the position, and a *U vector* and a *V vector* which define the orientation of the workplane. Only the directions of the vectors are relevant, i.e. using the vector (2,0,0) is the same as using (1,0,0). When calculating the *V* axis, the component of *V vector* parallel to *U vector* is removed to ensure that the two axes are orthogonal. Thus *V vector* may not be parallel to *U vector*.

The *Origin*, *U vector* and *V vector* fields may be specified using the mouse (see section 4.6.3), i.e. the workplane may be used to enter values which define the settings of the workplane itself. (Clicking on the view while any of these groups has focus will modify its values and move the cursor to the next group.) The *Apply* button updates the workplane without closing the dialog. Thereafter mouse point entry uses the new workplane.

The *Size* field controls the size of the rectangle and the grid drawn on screen. The workplane does not have to be symmetrical — it may, for example, extend from -1 to 3 in the *U* direction. It need not include the origin, i.e. it may also extend from -3 to -1 or from 1 to 4. For mouse interactions, the workplane extends to infinity, i.e. points can be selected outside the blue rectangle. While *Auto enlarge* is checked, the workplane will increase to include all geometry created while it is checked. This does not, however, resize the workplane relative to the existing geometry. That can be done by right clicking on the 3D view and selecting *Workplane → Fit to geometry*. Clicking the *Zoom to extents* button

on the 3D view toolbar also fits the workplane to the current geometry. The *Grid spacing* controls the density of the grid lines. If the workplane size is increased to the extent that the number of grid lines is more than 10 000 in either direction, the grid display button is automatically toggled off. The grid can still be displayed by activating this button again. However, if the number of lines exceeds 100 000 the grid is not displayed.

4.3.2 Transforming the workplane

The *Edit workplane* dialog defines the workplane in global coordinates. For arbitrary workplanes it can be very difficult to move the workplane normal to itself, or to rotate it around one of its own axes using this dialog. The *Transform workplane* dialog allows translation and/or rotation using the three workplane axes (U, V, N). The workplane settings are updated with the result of the transformation — any variables used in the settings will be lost. Select *Workplane* → *Transform* from the pop-up menu of the 3D view or click the *Transform workplane* button on the 3D view toolbar.

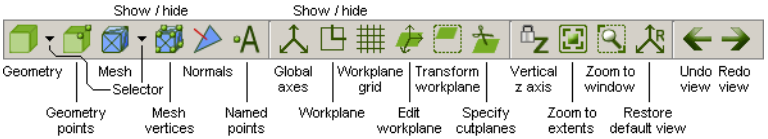
The rotation angles are in degrees. (In CADFEKO radian angles are only used in the arguments and results of the trigonometric functions and their inverses.) If rotation is done around more than one axis, CADFEKO uses Kardan angles. Simply put, this means the workplane is rotated first around the N axis, then around the V axis and finally around the U axis. (See section 9.2.38 for more details.)

The workplane cannot be transformed while the edit dialog is open, or edited while the transform dialog is open. The *Zoom to extents* operation (which fits the workplane to the geometry) is also disabled while editing the workplane.

4.3.3 Snap mode

Right click on the 3D view and select *Snap to* to specify how points will be selected from the 3D view via point entry (see section 4.6.3). If *Workplane* is selected, points are obtained from the workplane. If *Grid* is selected, the mechanism is the same, but the values are rounded to the nearest grid values. (The values are rounded in the workplane coordinates and may not appear rounded in global coordinates for arbitrary workplane orientations.) With *Geometry point* or *Mesh vertex* selected, the point is obtained from a point on the existing geometry or mesh. Selecting *Geometry face centre* or *Geometry edge centre* snaps to the centre of gravity of the face / edge under the cursor. For curved edges the geometric centre does not necessarily lie on the edge. When defining new geometry by snapping to existing geometry, the current coordinates of the existing geometry are used. There is no permanent relation and the new point is not updated when the existing geometry is changed. The objects can be modified together if they are defined using named points (see section 4.5.1). This is done by selecting *Snap to* → *Named point.*

4.3.4 The 3D view toolbar



The first nine toggle buttons on the toolbar at the top of the 3D view control the visibility of objects. Geometry (solids, surfaces and lines), geometry points, mesh elements, mesh vertices, mesh element normals, named points, the global axes, the workplane and the grid can be shown or hidden with the respective toggle buttons. (If the button is toggled on (down) the corresponding items are shown.) Hiding the workplane also hides the grid, but not the other way around. The *View mesh edges* button on the main toolbar (see section 4.11) also controls element visibility, but it is relevant to the selected items and applies to all 3D views simultaneously. Showing / hiding individual items (see section 4.3.6) overrides the settings on the 3D view toolbar.

The down arrows next to the *Geometry* and *Mesh* view buttons select the type of view. In the *Default* view, all parts are drawn with solid faces and the same colour.

If *Colour by medium* is selected, each medium is displayed with a separate colour as indicated in the list under *Media* in the tree (see section 4.8). Surface mesh elements are coloured on each side according to the medium on that side of the face. For example, when viewing the mesh of a dielectric object, the entire object will have the free space colour when viewed from outside. The different colours of the internal media will only become visible if cutplanes are introduced. (Since the user can set arbitrary colours on each side of a triangle, this colouring scheme is required to verify that the media was specified correctly.) The same applies to geometry faces between different dielectic media. External geometry faces are, however, displayed using the colour of the internal medium. Hence a dielectric object will be displayed with the dielectric colour also when viewed from outside.

If *Wire-frame* is selected, only the edges of the model is shown. This allows viewing hidden objects, but can be a very confusing display. Since only the edges are displayed, cylinders will look the same as two disks and spheres are completely invisible.

The *Edit workplane*, *Transform workplane* and *Specify cutplanes* buttons provide quick access to these functions. The *Vertical Z axis* button controls the rotation of the model in such a way that the Z axis remains vertical on the screen (i.e. locking it in place). The next two buttons zoom to the full model extent or to a selected window, respectively. (To zoom to a window after clicking this button, click and drag the mouse, without releasing the button, to define the window.) The *Restore default view* button resets the view angles to the standard position.

The *Undo view action* and *Redo view action* buttons (also accessed with the short-cut keys <Alt><←> and <Alt><→>) apply to all view manipulations in the current view. View manipulations are remembered in a separate list from the one used to store model manipulations, so that view operations can be undone separately from geometry modifications. The *Undo* / *Redo* buttons are disabled if there are no additional undo / redo operations.

4.3.5 Cutplanes

Right click on the 3D view and select *Cutplanes* to open the cutplane dialog for that view (or click the *Specify cutplanes* toolbar button). The first cutplane is created automatically when the dialog is opened. The *Add* button creates additional cutplanes. The *Remove* button deletes the current cutplane, while unchecking the *Active* item deactivates the cutplane without losing all its settings. The *Flip* button hides items on the other side of the cutplane.

The cutplane is specified similar to the workplane (see section 4.3.1), using an origin and two vectors. These fields all use the standard point entry (see section 4.6.3). Thus the cutplane can be interactively modified. The plane preview is shown in all 3D windows and accepts point entry from any window, but it applies only to the current window.

The cutplane is updated with each change in the dialog. Closing the dialog with *Cancel* reverts to the state when the *Apply* or *OK* button was last clicked.

Where a cutplane passes through solids, the cut surface is displayed in a slightly different colour. These surfaces cannot be selected, thus clicking on them will select geometry that lies behind the cutplane. These surfaces are only shown when the current selection (see section 4.11) mode is geometry parts. When selecting anything other than geometry parts, CADFEKO cuts solids similar to shells (showing only the actual faces) allowing access to the internal faces and edges.

The cutplane icon on the right of the status bar is active if the view under the mouse cursor has active cutplanes. Like the snap options, this applies to the view under the mouse cursor rather than the active view.

4.3.6 Selective viewing

In some cases where geometry or mesh parts are obscured by other items it can be difficult to set up cutplanes to view these items. It is therefore possible to selectively hide geometry and mesh parts. Hidden items are removed from all 3D views, but are still part of the model and are still meshed and exported to the CFM file. (Meshing a hidden object will result in a visible mesh part — even if a hidden mesh part with that name existed beforehand.) Hidden items are displayed with grey icons in the tree.

Items are hidden by selecting them and selecting *Hide* from the pop-up menu or *View* → *Hide* from the main menu. These items can be shown again by selecting them in the tree and selecting *Show* from the pop-up menu or *View* → *Show* from the main menu. If the selection contains visible and hidden items, selecting *Show / Hide* toggles the hidden state of each selected item. Finally, selecting *View* → *Show all* in the main menu shows all hidden items independent of the current selection and the pop-up menus *Show all geometry* and *Show all meshes* do the same for geometry and mesh parts respectively.

If a model is saved and reloaded, all items become visible again.

4.4 The tree

The tree provides an alternative representation of the current model. The top (or contents) tree contains the lists of variables, named points and media, and an overview of the geometry and meshes.

The entire history of the geometry creation can be seen in this tree. Where objects are derived from existing ones (for example, the individual objects used in a Boolean operation or the original object in a split operation), the original (parent) objects are removed from the top level of the model and listed under the new object in the tree. The term *part* is used for highest-level items. These can be at the root level under *Geometry* or at the top level in an assembly and are the objects that are visible in the 3D view.

The tree can be expanded / collapsed from the selection downwards by pressing `*` or selecting *Collapse / Expand all* from the pop-up menu. This applies to the focused item in the contents tree — independent of which items are selected. (Unless detail — for example faces — are selected, the focused item will be part of the selection.) Collapsing an entry will show all the direct parents only. To hide the direct parents (but keep the rest of the state) use the `-` key. Likewise the `+` key expands only the current level.

When specific items are hidden (see section 4.3.6) they are displayed with gray icons in the tree.

The bottom (or details) tree shows the detailed information (faces, edges, regions, transforms, etc.) of a single part, displayed in the contents tree with a bold italic font. The lists of faces, edges and regions only apply to top-level parts — these items are not listed for parent objects. Icons next to the items in the trees indicate their type, properties and the application of transforms, mesh properties, etc. (The default names also indicate the type, but after renaming, for example, `Assembly1` to `Antenna`, the icons become very useful.) The square icons next to dielectric names and regions indicate the colour of this medium in the 3D display.

Primitives	
	Cuboid
	Flare (cut-off pyramid)

	Sphere
	Cylinder
	Cone
	Polygon
	Ellipse
	Parabolic surface
	Line
	Polyline curve
	Beziér curve
	Elliptical arc
	Spline curve through points
	Helix or spiral
	Imported CAD body
	Surface body (e.g. created with face copy or explode)
	Curve (edge / wire) body (e.g. created with edge copy or explode)
	Mesh part

New constructions	
	Boolean union
	Boolean intersection
	Boolean subtraction
	Split
	Object created by spinning / rotating parent
	Object created by sweeping / extruding parent
	Loft (ruled surface) created between two edges
	This geometry contains edges created by projecting other items onto it
	This geometry contains imprinted points
≈	Simplified geometry (redundant items removed)
	This is an assembly

Indicators	
	This item has been transformed
	This part contains dielectric regions
	Metal face on a dielectric
	Non-metal face on a dielectric
?	This item is suspect (it could not be mapped)
	Local mesh properties (on regions, faces, edges)
	Local wire radius

Transforms	
	Translation
	Rotation
	Scale
	Mirror

Right clicking in the tree will open an appropriate pop-up menu. Double clicking on an item in the tree will display the *Properties* for that item, except for the *Variables* and *Named points* headings, in which case a new item is added).

The tree area can be hidden by dragging the slider all the way to the left and shown again by dragging the slider to the right again or by selecting *View* → *Tree* from the CADFEKO menu.

4.5 Variables and named points

CADFEKO supports parametric models. Thus most input values can be specified using variables or mathematical expressions such as `1+sqrt(x)`, where `x` is a user defined variable. The expressions are then stored with the model and, when a variable is changed, all items depending on it are updated. It is therefore very easy to adapt a model using variables and a little planning, for example, to modify the geometry for use at a different frequency. Since it is possible to edit geometry objects, variables can be introduced after constructing the model.

While the geometry is fully parametric, the mesh is not, because storing expressions for mesh vertices would have severe implications on memory requirements. In general, operations working at the part level are fully parametric, but operations that apply to individual elements are not. When working with mesh elements, most input fields still accept expressions. For example, just add `+1` to a coordinate of a vertex to move it by one unit. These expressions are then converted to values when the operation is executed.

Variables may be added by double clicking on the *Variables* entry in the tree or selecting *Geometry* → *Add variable* from the main menu. The first character of a variable name must be alphabetic — 'a' though 'z' and 'A' through 'Z' — or the underscore; the remaining characters may also be numeric — '0' through '9'. Variable names are case insensitive, i.e. 'a' and 'A' are treated as the same character. The *Expression* defines the value of the variable and may be a simple number (such as `1.23`) or a mathematical expression which may use round brackets, the operators `+`, `-`, `*`, \, `^` (exponential notation), other variables, as well as the functions listed below.

fmod	`fmod(a,b)` returns the remainder of the division *a/b*
sin cos tan cot	trigonometric functions (arguments in radians)
arcsin arccos arctan arccot	trigonometric inverse functions (results in radians)
atan2	`atan2(y,x)` yields arctan*(y/x)* in the range $-\pi \ldots \pi$
deg	converts radians to degrees
rad	converts degrees to radians
sinh cosh tanh	hyperbolic functions
log	logarithm to base 10
ln	natural logarithm
exp	exponential function
sqrt	square root
abs	absolute value
step	`step(x)` is 1 when $x > 0$; otherwise it is 0
ceil	rounded upwards
floor	rounded downwards
min max	`min(a,b)` or `max(a,b)` gives the minimum / maximum of the two arguments

If there is an error in the expression, an error message will be displayed at the bottom of the dialog. The variable can only be added when the expression is valid. The *Evaluate* button evaluates and tests the validity of the expression without closing the dialog. The result is maintained until the next time the expression is evaluated — it is not updated automatically when the expression is changed.

Variables can be changed by double clicking on the variable itself (or by right clicking and selecting *Properties*). CADFEKO will display a *circular dependencies* error if a variable is changed in such a way that it depends on itself. All items depending on the variable will be updated automatically. If this results in invalid geometry (for example, a sphere radius that becomes zero or an intersection or split that becomes empty), an error is written to the message window and the variable is not modified. The *Modify variable* dialog may be left open while editing the item which gave the error. Thereafter focus reverts back to the *Modify variable* dialog. Variables can be deleted via the pop-up menu. The delete operation will fail for all variables that are still used (for example, in defining other variables or geometry). The variables

c0	the free space wavelength
eps0	the free space permittivity
mu0	the free space permeability
zf0	the intrinsic wave impedance of free space
pi	the constant π

are defined as part of a new model. There is nothing special about these variables; they may be deleted and/or modified just like any other variable.

4.5.1 Named points

Named points can be created by double clicking on the *Named points* entry in the tree, by right clicking it and selecting *Add named point*, or by selecting *Geometry* → *Add named point* from the main menu. They are effectively vector variables and are typically used to create geometry that should change when the point is modified (see section 4.6.3). The *Add named point* dialog contains input fields for the three coordinates of the point in the global coordinate system.

The x, y and z components of a point are accessed using a dot followed by the required component, for example, 'P.x' is the x component of point P. Points can be constructed with the 'pt' command, for example, 'pt(1,1,1)'. Only the subtract and add operations are allowed between two points (the corresponding components are added / subtracted) and a point may be multiplied with or divided by a scalar. In addition, the 'abs' function listed above may be applied to points. It gives the distance from the origin to the point, i.e. the vector length if the point is interpreted as a vector. Currently no other operations may be applied to points.

4.5.2 Calculator

Select *View* → *Calculator* to open the *Calculator* which allows evaluating and testing variable and named point expressions without modifying any model entries.

The format of the result can be controlled. *Scientific* uses exponential notation, for example, `0.01` becomes `1.0e-2`. *Engineering* format is similar to *Scientific*, except that the exponent part is always a multiple of 3. Thus `0.01` becomes `10.0e-3`. *Decimal* uses a fixed notation without exponent. This is not recommended for small numbers, for example, with 5 decimals, `1.0e-6` becomes `0.00000` which means all information is lost. The *Decimals* field gives the number of digits after the decimal point.

4.6 Creating geometry

4.6.1 Geometry extents

Operations on geometry (such as checking if two points are the same) require a numeric tolerance. This tolerance is dependent on the model size. (For example, microstrip structures could require defining dimensions within a few micrometers, but this accuracy is clearly not needed when doing propagation studies around mountains.) CADFEKO therefore allows the specification of the model size — select *Geometry* → *Extents* from the main menu. The extent is the same in all directions. Thus only one value needs to be specified. This is the *Maximum coordinate* which gives the largest offset — in either direction — along any of the three axes. For example, if the *Maximum coordinate* is 500, the entire geometry must fit inside a 1000x1000x1000 unit box centred at the origin. (Here "unit" refers to the numbers as used in CADFEKO — see section 4.16.1.)

The model tolerance is then determined as the *Maximum coordinate* divided by 5×10^8. Distances shorter than this are assumed to be zero.

The default setting is `5E+02` and it is recommend that this be used, unless the model is larger (for example, when modelling an automobile in mm), or much smaller (requiring a more accurate model and/or mesh). For settings other than `5E+02`, exported Parasolid models will not be in the same units used in CADFEKO. (See section 4.6.7.)

The size box applies to all geometry. For example, an intersection between two spheres cannot be created if either sphere exceeds the size box even if the result is inside it. Note that if the geometry exceeds the specified size, CADFEKO will display an error. The extents can then be changed without having to close the dialog that caused the error.

4.6.2 Local coordinates

All primitive geometry objects in CADFEKO are defined in global or local coordinates. The default coordinate system can be specified by selecting *Options* → *Preferences* from the main CADFEKO menu, but it can be changed on each geometry dialog. If the *Use global coordinates* box on the creation dialog is unchecked, the *Local* tab as shown in figure 4-4 becomes active. Specification of the local coordinates is very similar to specifying workplanes (see section 4.3.1). The *Origin* field specifies the origin of the local coordinates. Changing this value will translate the coordinates and thus the position where the new object will be created. The *U vector* and *V vector* fields define the directions of the U and V axes and control the orientation of the object. The third axis N is normal to the U and V axes. The *Set to workplane* button sets the local coordinates to that of the workplane in the current view. Thus the workplane tools can be used to define and manipulate the workplane, and then the local coordinates fitted to this.

If the preference is set to local coordinates, the *Use global coordinates* box is unchecked and the local coordinates initialised to the workplane of the active view each time a

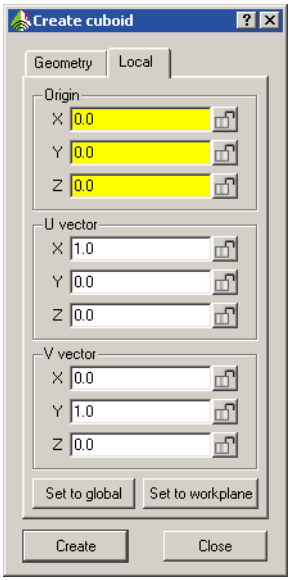


Figure 4-4: Local coordinate panel of the *Create cuboid* dialog

geometry creation dialog is opened. (This is only done when opening the dialog — the coordinates are not modified if the workplane is modified after opening the dialog.) If no 3D view exists, the local coordinates are matched to the global coordinates.

4.6.3 Point entry

Some fields on the geometry creation dialogs (amongst others) allow entering values by clicking with the mouse on the 3D view. If such fields have focus, their background colour changes to yellow. By clicking on the 3D view, the coordinates of the selected point (determined using the active snap mode — see section 4.3.3 — of the particular view) are entered into the field and the focus shifted to the next field. It is also possible to click on a named point or variable in the tree. This point is then added in all the active fields or, in the case of a variable, the current field. This allows creating geometry with a series of clicks on the 3D view and one click on *Create* in the dialog. For one-dimensional input fields (such as the radius of the sphere), the value is calculated from the specified point and the result entered into the field.

Where the input field requires a vector, the vector is considered to lie from the origin to the selected point. It is important to realise that the sequence of operations can have a significant effect. For example, if the *U vector* of the workplane is defined by snapping to some model point (such as a mesh vertex), the vector from the origin to this point is calculated and entered into the field as a numeric value. If the origin is modified

afterwards, the direction of the *U vector* remains unchanged. If, however, the direction of the vector is specified by snapping to a named point, it is re-evaluated whenever the origin is changed. The *U vector* then keeps pointing at the specified point.

Fields which accept multiple values from point entry have *Lock* buttons next to them. If such a button is toggled on (down), that field is not updated when selecting a point. This is used when there is a need, for example, to click on points in the workplane without modifying the value in the *N* direction. It is also possible to specify different named points in the three component fields of a point. Each component is then determined by projecting the specified point along the required dimension.

As values are entered into the fields, an outline preview of the geometry is shown. If <Ctrl><Shift> is pressed while moving the mouse over the screen (without clicking), the fields values are updated continuously. This is called *Preview mode*. The active fields show the values that would be entered if the mouse was clicked at that position. The values in the active fields are displayed using an italic font to indicate that preview mode is active. The snap mode and the snapped coordinate position (in global coordinates) are shown on the status bar. If <Ctrl><Shift> is released without clicking on the view, the values are reset to what they were before preview mode was activated. When editing primitives that have been transformed, a dashed outline is drawn at the original position in addition to the solid outline at the final position.

Some dimensions may be orthogonal to the workplane. (For example, the height of a cylinder when the local coordinates are aligned with the workplane.) When such values are active while entering points with snapping to the workplane or grid, the height is determined by projecting the point to a line normal to the workplane. This is only possible when the view direction is not orthogonal to the workplane. The view orientation of the model can be changed by left clicking and dragging the mouse (as opposed to just clicking) — even while creating geometry.

When clicking the *Create* button, all input fields are evaluated. If any field contains an invalid expression, or a value that is not valid for that parameter (such as zero for the radius of a sphere), an error is displayed in the message window and the field indicated.

4.6.4 Creating solid primitives

The *Geometry* → *Solid* menu contains sub-menus to create new cuboids, flares (a possibly truncated pyramid), spheres, cylinders or cones. The entry process is very much the same for all primitives, so this discussion deals primarily with cuboids.

The *Create cuboid* dialog — shown in figure 4-5 — allows creating cuboids. It requires the position of one corner and the dimensions along the three coordinate directions. Negative values are drawn in the negative axis direction.

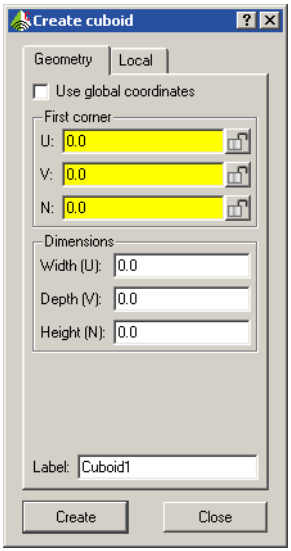


Figure 4-5: The geometry panel of the *Create cuboid* dialog

A new object is created each time the *Create* button is clicked. When all the required cuboids have been created, click *Close* or press <Esc> to close the dialog. The dimensions must be non-zero. If any dimension is zero, CADFEKO will display an error in the message window and indicate the offending field. The object will not be created. Where allowed (such as the width of a cuboid), negative values are taken in the opposite direction.

The *Label* field allows entering the name of the new object. This is visible in the tree and can be changed at any time.

If the *Radius* field for a sphere or cylinder is specified with a point, the surface of the sphere or the extended cylinder will pass through that point. For a cone the radius is determined as the distance between the origin of the local coordinate system and the projection of the point in the local *UV* plane. To specify a sharp tipped cone, leave the *Top radius* field `0`.

New solids are metal by default, but can be changed to dielectric or shell structures. This is done by setting region properties (see section 4.8).

4.6.5 Creating surface primitives

All 2D surfaces are created as solid sheets. They therefore also contain a *Face* entry in the tree. (In this document surfaces refer to parts or components. Faces refer to individual sheets making up the surface or defining the boundary of a solid.) Currently only flat polygons, elliptical disks and paraboloid surfaces are supported.

Polygons are created with the dialog shown in figure 4-6. All the specified points must lie in a plane — the simplest option is to align the local coordinates with the plane of the polygon and ensure that all N coordinate values are zero.

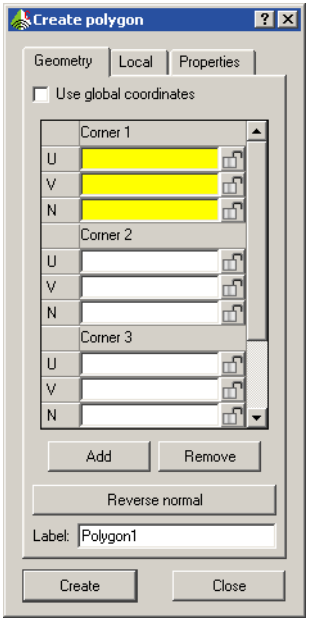


Figure 4-6: The dialog used to create polygons

Clicking *Add* adds an additional corner after the one that has focus or, if no corner has focus, at the end of the list. The polygon is created by connecting the lines in the specified order and the edges are not allowed to cross. The corner with focus is shown with a blue square in the 3D display. If the last coordinate is entered with the mouse, CADFEKO automatically adds another point and moves the focus to it. The *Reverse normal* button reorders the points in such a way that the normal vector (determined in a mathematically positive sense from the direction of the edge) is reversed. When *Create* is clicked, all empty points are removed automatically. For polygons, the *Properties* tab contains an entry *Unmeshed plate*. If this is set to true, the polygon is meshed into a single polygonal plate rather than into triangles. This is then used with the UTD in FEKO.

The *Label* field allows entering the name of the new object. This is visible in the tree and can be changed at any time.

The radii fields for the ellipse and the paraboloid primitives must be positive.

The paraboloid is oriented along the N axis as shown in figure 4-7. The *Radius* field specifies the radius R in the plane orthogonal to the axis and must be positive. The *Focal depth* is the distance f from the origin to the focal point. If this is negative, the paraboloid is oriented along the -N axis. The focal depth is related to the height h by

$$f = \frac{R^2}{4h}.$$

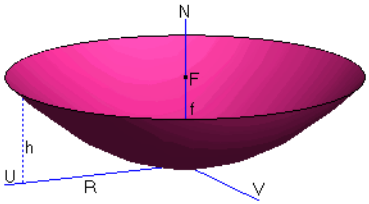


Figure 4-7: Dimensions of a paraboloid

4.6.6 Creating curve primitives

Curve parts in CADFEKO can be used either as building blocks for constructing geometry (for example, using the spin, sweep or loft operations) or as free-standing conducting wires. In addition to the general curves resulting from intersections between faces of shell bodies, CADFEKO supports a number of simple primitives. These include single straight lines, polylines, fitted splines, Beziér curves, elliptical arcs and helices / spirals. The creation of curve primitives is very similar to the creation of solids and surfaces.

The polyline and fitted spline primitives make use of a list of points. Here, as for polygon surfaces, new fields are created if the last field is entered with the mouse, and blank points are removed before constructing the geometry. The helix component can also create conical spirals by setting the top and bottom radii to different values, or flat spirals by setting the height to zero.

Note that the wire radius (see section 4.12.1) is specified on an edge (as listed in the details tree) rather than on a primitive. Specifying the radius for a curve created from the intersection of faces is done in the same way.

4.6.7 Importing / exporting geometry

CADFEKO is built on the Parasolid solid modelling kernel. Hence Parasolid models can be imported and exported without any translation.

Select *File* → *Export* → *Parasolid...* to export the geometry to a Parasolid file. This opens an *Export Parasolid model* dialog — where the choice between *Text* or *Binary* format must be made — followed by a dialog prompting for the name of the file to export to. CADFEKO currently exports Parasolid version 16.1, but with the correct schema file this can also be imported into older versions of Parasolid. Only the final geometry is exported. Exporting and importing the same model looses the entire creation tree.

Parasolid models are inherently limited to a 1000x1000x1000 unit box centred at the origin. CADFEKO introduces a scaling factor to counteract this (see section 4.6.1). The *Scale factor* field on the *Export Parasolid model* dialog displays the factor by which the model in CADFEKO units must be multiplied to convert it to the units in the Parasolid model. A scale factor of `0.1` implies that the dimensions of the saved Parasolid model are one tenth of the native dimensions as used in CADFEKO. Typically, programs that import Parasolid models allow specifying a factor with which the Parasolid model must be multiplied during the import. This must then be the inverse of the scale factor reported by CADFEKO. For convenience, the import factor is stored in a separate file (the filename is derived by adding `.info` to the full Parasolid filename, for example `export_geometry.x_t.info`) in the same directory as the Parasolid file.

If the model needs to grow larger than 500 units, the extents must be increased. If, however, the model is smaller than 50 units, the extents can be decreased. This is not recommended (unless the model is very small and runs into precision problems), since using the default extents results in an unscaled Parasolid model, removing the need to keep track of the scale factor.

Parasolid models can also be imported into CADFEKO. Select *File* → *Import* → *Geometry* → *Parasolid...* from the main menu. This asks for a filename and then opens an *Import Parasolid model* dialog where the scale factor can be modified. This *Scale by* factor is the factor by which the Parasolid model is multiplied as it is imported. If the Parasolid model was created by CADFEKO and the INFO file is available, the dialog will display the value in the file. This value can then be changed. For example, if an antenna model was constructed in mm and then imported into an automobile model constructed in m, a scale factor of 0.001 would be appropriate (independent of any scale factor required by the definition of the model extents.) Note that importing a large model with a scale factor other than unity can be a very slow process.

Isolated vertices (acorns) in a Parasolid model are not the same as named points (see section 4.5.1) in CADFEKO. Hence they are not imported, but CADFEKO writes their coordinates to the message window. The required named points can then be created manually.

If *Import bodies with faults* are checked, all parts in the file are imported — even if they contain errors. It may then be possible to, for example, explode such parts and delete only a few problematic surfaces. However, this option should be used with care — it allows adding possibly corrupted data into a model which is assumed to be consistent. The model should be saved, preferably using different names, before and after this operation.

CADFEKO can also import other CAD formats using the Spatial InterOp translators. Select *File* → *Import* → *Geometry* → ... from the main menu. (Note that the translators are licensed separately. If some menu items are disabled, you do not have a licence to use these import modules. If you need or want to evaluate these modules, please contact your distributor.)

If an import menu is selected, CADFEKO asks for a filename and then opens the *Import ... model* dialog. The *Units* group on this dialog displays the units of the imported model and allows setting the *Destination* units (which should be set to the unit of the numbers as used in CADFEKO). These units controls the scaling of the imported model.

Since the imported model is converted to Parasolid, all the comments for Parasolid imports also applies here. Due to the differences in the internal representations, converting CAD formats can have additional unexpected results. In particular there is often situations adjoining surfaces no longer line up correctly. This problem is compounded by the fact that CADFEKO does not yet have any CAD fixing tools. Some imported models use a numerical representation and scaling these can lead to problems.

The conversion is logged in the file `%FEKO_HOME%\logs\CADimport.log` which is especially useful in cases where the conversion fails.

4.7 Operations on geometry

Any combination of objects can be selected and modified. Most options are available from the main menu or the *Modify geometry* toolbar. Modify operations include transformations, Boolean operations (*Union*, *Subtract* and *Intersect*), other advanced operations on parts (such as projections, *Split*, *Spin* and *Explode*), *Rename* (to modify the label / name of an object or one of its regions / faces / edges — all objects in the model must have locally unique names), *Copy* (discussed below), *Delete* (also available via the short-cut key <Del>), *Reverse face normals* and *Properties* (which opens an edit dialog similar to the create dialog.)

Complex geometry is created from a sequence of operations starting with simple primitives. This sequence is shown in the tree. It is possible to select and modify objects at any level in this tree, or change variables upon which the geometry depends. The tree is then re-evaluated. During this process CADFEKO may not be able to maintain the identities of some regions / faces / edges, for example, where multiple faces are derived from the same original face during Boolean operations. These items are then marked *suspect* (see section 4.4) as a warning that properties set on them may not have been maintained. After ensuring that the properties are correct or have been corrected, the faces may be reset using *Set not suspect* from the pop-up menu.

The parametric capability in CADFEKO (see section 4.5) makes it possible to edit the components of complex geometry operations at any time. In most cases it may be quite difficult to identify which component to change. CADFEKO therefore displays a wire-frame representation of all selected parent items.

4.7.1 Transformations

Geometry objects can be transformed by selecting them and selecting *Edit* → *Transform* from the main menu, or using the transform buttons on the *Modify geometry* toolbar. The *Transforms* entry in the details tree contains a list (in creation order) of all the operations applied to each object. These may be edited (by double clicking on the specific operation) or deleted (by pressing the <Del> key). A transform may be applied to a selection of multiple objects, but it will be added separately to each object. Therefore, changing them afterwards requires selecting and editing the transform of each object separately. The transform can, of course, be defined in terms of variables. In this case, changing the variable will modify all the items.

The *Rotate* operation requires an axis of rotation (which is determined by an origin and an axis direction, both in global coordinates) and a rotation angle (in degrees). The angle is measured in a mathematically positive sense around the axis direction. The dialog, shown in figure 4-8, also provides buttons to set the rotation axis to the principal axes of the global and local coordinate systems. The rotation is applied to the object, not its local coordinates.

The *Translate* dialog, shown in figure 4-9, requires a start and end point from which it calculates the translation distance and direction.

The *Mirror* operation requires a plane. This is specified via the dialog shown in figure 4-10 and is similar to the specification of the workplane. The dialog also provides buttons to select from a number of common planes.

The *Scale* transformation requires an origin (in global coordinates) and a scale factor as shown in figure 4-11. Scaling is done around the origin specified on the scaling dialog, not around the centre or origin of the object or the model.

4.7.2 Multiple (transformed) copies

Parts can be copied and transformed multiple times via a single command. Select *Copy special* from the pop-up menu or *Edit* → *Copy special* from the main menu. The transform dialog then contains the *Number of copies* field shown in figure 4-12.

This will make the specified number of copies of each selected part. With rotation the n^{th} new part will be rotated (with a single transformation) through n times the specified rotation angle. For translation the n^{th} new part will be translated by n times the specified translation distance. For the mirror operation only one copy is allowed. (A second copy would be back on the original and can be done with the standard *Copy* option.)

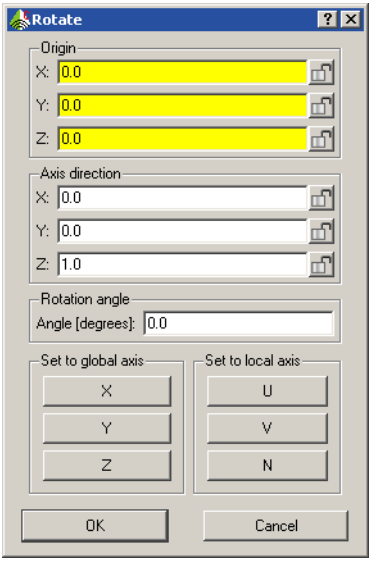


Figure 4-8: The rotation dialog

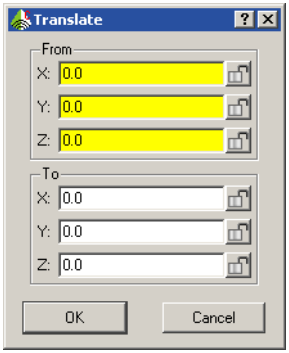


Figure 4-9: The translation dialog

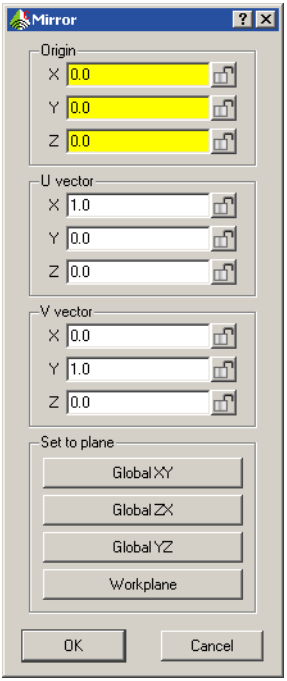


Figure 4-10: The mirror dialog

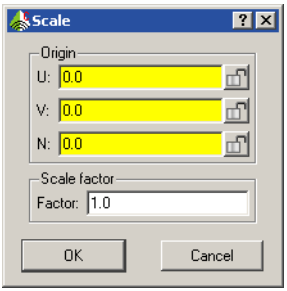


Figure 4-11: The scale dialog

Figure 4-12: The multiple copies section of the transformation windows

4.7.3 Boolean operations

The *Geometry* main menu and the *Geometry modification* toolbar provide access to the Boolean operations *Union*, *Subtract* and *Intersect*. The parts must be selected before requesting a Boolean operation. *Union* combines all selected parts. For *Intersect*, the result is the common part of all the selected parts. For *Subtract*, all the selected parts are subtracted from a final part, which must be selected when prompted for. When only one part is selected, the only Boolean operation allowed is *Subtract*. No Boolean operation is available when there are no selected parts. (Surfaces and wire bodies are also parts.) Boolean operations cannot be applied to parents or the regions / faces / edges of parts.

4.7.4 Split parts

Selecting *Geometry* → *Split* from the main menu will open a dialog where the split plane can be specified similar to the way the workplane is defined. The split plane defaults to the workplane of the active view. This operation creates two new parts (named *Split_back...* and *Split_front...*, where the *Front* side of the split is in the direction of the positive N axis of the split plane) for each selected part. The two halves of each split are derived from independent copies of the original part. Thus if the entry under *Components* is modified for one half, the other half remains unchanged and has to be modified separately. If synchronised changing is required, the original part must be created using variables. The split plane may be modified by selecting the new part and selecting *Properties* from the pop-up menu.

4.7.5 Spinning and sweeping parts

Selecting *Geometry* → *Sweep* or *Geometry* → *Spin* from the main menu (also available on the *Modify geometry* toolbar) sweeps (also referred to as extrudes) the selected parts along a specified vector or spins (rotates) them around a specified axis, respectively. These operations can only be applied to parts. The sweep / spin operation is applied separately to each selected parts. If multiple parts are swept / spun, the resulting new parts have no relation to each other, i.e. each one gets its own copy of the parameters and can be modified independently afterwards.

Only parts containing nothing but edges and/or faces (not solids or closed regions) may be spun / swept. For surface bodies, the body must have a single boundary which does not close on itself and no edge may be attached to more than two faces (for example, the T-plate union in figure 4-13 is not allowed to be spun / swept as the connecting edge is bordering three faces). Sweeping or spinning an edge results in a surface, while applying these operations to a face results in a solid.

When the *Sweep* operation is selected, the *Sweep geometry* dialog allows specifying the vector in terms of a start and end point. These fields accept standard point entry (see

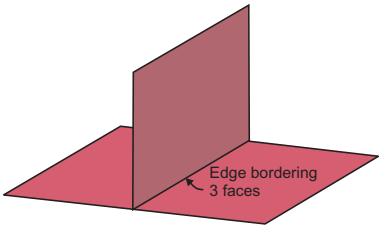


Figure 4-13: T-plate example of an edge bordering three faces which means that it cannot be swept / spun

section 4.6.3). It is not possible to sweep bodies in a direction that is tangential to any of its edges or in the plane of its faces.

The *Spin geometry* dialog allows specifying the axis of rotation (an origin and a direction) and a rotation angle. The angle is taken in the mathematically positive sense around the specified axis and is specified in degrees. (In CADFEKO radian angles are only used in the arguments and results of the trigonometric functions and their inverses.) The *Set to ... axis* buttons allow quick selection of common axes. Note that the local axis buttons are only available if all selected parts have the same local coordinate system.

Note that the normal directions of the resulting surfaces depend on the direction of the curves.

There are a number of restrictions on the relation between the part and the spin axis. No free edges may be coincident with the axis, nor are they allowed to intersect the axis at any point other than at the ends of the curve. For full spins of parts with faces, the axis may not intersect the face at only one point, unless it is at an end point of an edge. The axis may be coincident with any edge of the sheet, provided that the entire edge lies on the axis. In addition, no edge may touch the axis tangentially as shown in figure 4-14 — not even if the edge is broken at this point. (Again, it is allowed if the entire edge lies on the axis.)

4.7.6 Loft surfaces

The *Loft* operation forms a smooth surface by connecting two curve parts with straight lines. This is sometimes called a ruled surface, but it is not a faceted model — unless one or both curves are polylines. Curve bodies may consist of a number of edges, for example, a polyline. The two lofted curves must have the same number of edges (the loft can be considered to connect each pair of edges). If this is not the case, points can be imprinted (see section 4.7.7) on one of them to rectify this. Curves with more than two edges joining at one point are not allowed. The preview shows how the curves will be connected. The *Loft* dialog allows reversing the start and end points of one edge in the case where the two edges are not created in the same (or desired) direction.

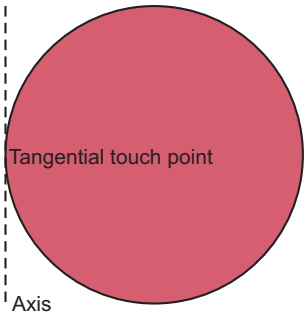


Figure 4-14: Example of a surface with an edge that just touches the axis of rotation

4.7.7 Projection and imprinting points

In some cases it is desirable to create specific points, edges or faces on a given geometry, for example, to allow attaching other structures to the model, to specify a finer mesh size along a curve or to create metallic patches on dielectric objects.

The *Imprint points* operation allows placing specified points on the selected part. Points can only be imprinted on one part at a time. The *Imprint points* dialog allows a list of points to be specified using standard point entry (see section 4.6.3) in either global or local coordinates. The specified points are projected onto the closest point on the selected part — either on a face or an edge. Points may not be imprinted on existing points. The imprint operation creates a new entry in the tree (allowing access to the part without points), but uses the name of the parent object, since there is only one parent.

Multiple parts can be projected onto another part by selecting them, activating the *Project* operation and then selecting a target part to project onto. This sequence is similar to the subtract operation. Also similar to the subtract operation, all the projected parts appear as parents of the resulting part, i.e. they are no longer present as individual parts of the model. All the edges of the selected parts are projected onto the faces of the target part. Any part (curves, surfaces and solids) can be projected onto any part containing faces. (Spheres do not have edges, hence projecting a sphere has no effect other than removing it from the model.) Where the projected edges form closed paths, new faces are created.

The projection direction is determined from the normals of the faces of the target. Projecting edges onto convex curved target faces will thus tend to reduce their size and/or perspective. Convex surfaces may also "shadow" other surfaces in that all points on the projected edge may project onto the curved face even though it seems as if it should also project unto another face. Finally, edges are only projected onto the normal side of faces.

Edges where the projection crosses itself or turns back on itself are not allowed.

4.7.8 Reversing face normals

Face normals can be reversed by selecting the faces and selecting *Reverse face normals* from the pop-up menu in the details tree or *Geometry → Reverse face normals* from the main menu. (The normals of all selected faces are reversed, even though only one part's faces are shown in the details tree at any given time.)

Currently, it is not possible to reverse the normal on a body with a single, closed face, for example, a sphere. If such a normal must be reversed, the simplest option is to project a circular line onto the body such that it creates two faces. The normals of these faces can then be reversed and the edge deleted by simplifying (see below) the part afterwards.

4.7.9 Removing detail and the simplify operation

Redundant faces may be deleted by selecting them and pressing the <Del> key or selecting *Delete* from the pop-up menu. Faces can only be redundant if they have the same medium (i.e. metal, free space or the same dielectric medium — see section 4.8) on both sides. When a face separating an internal free space region from the outside free space is deleted, the internal region is merged with the outside. Since the outside medium is free space, surfaces can only be removed from closed regions if the internal medium is set to *Free space*.

Redundant and free edges may be deleted similarly. These items are permanently removed, but it is possible to copy the original object (see section 4.7.11). Also, if a component or variable is changed in such a way that the object is re-evaluated, suspect faces (see section 4.7) will not be removed and some deleted faces may reappear. Note that edges are not redundant if the normals of the surfaces are in opposite directions. For this reason the back sides of faces are coloured different that the front (normal) sides in the *Default* view and it is possible to reverse face normals (see section 4.7.8). The normals of triangles are in the same direction as the surfaces they belong to.

CADFEKO also allows automatic removal of redundant faces and edges. Select *Geometry → Simplify* to open the *Simplify geometry* dialog.

This operation allows removing specific types of items. The simplified geometry will be electromagnetically the same as the original, but may not have the same meshing constraints. If, for example, an imprinted point is removed, there will no longer be a guaranteed mesh vertex at this location. Faces cannot be deleted unless the regions they separate can be merged. The same applies to edges on the boundaries of faces and geometry points at the ends of edges.

By default the simplify operation does not remove redundant regions, faces or edges on which local mesh properties (see section 4.12.1) are set. To remove them, the various *Keep ... with local properties* options must be unchecked. For example, consider the union of two spheres of a similar dielectric medium shown in (a) in figure 4-16. A local mesh size is

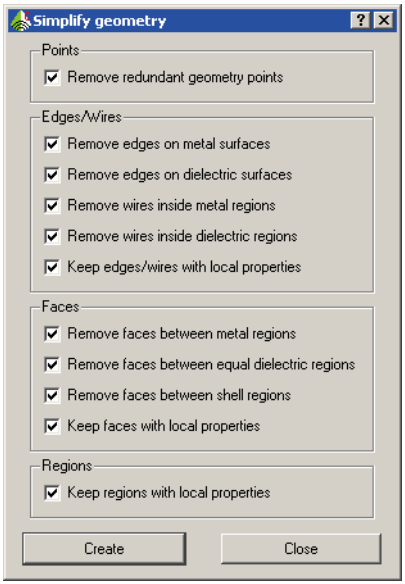


Figure 4-15: The simplify dialog

set on *Region1*. If the union is simplified with *Keep regions with local properties* checked, the result is as shown in (b). Since the region contains local properties, it is not removed. Then the face between it and the centre region is also not removed. If *Keep regions with local properties* is unchecked, the result is as shown in (c).

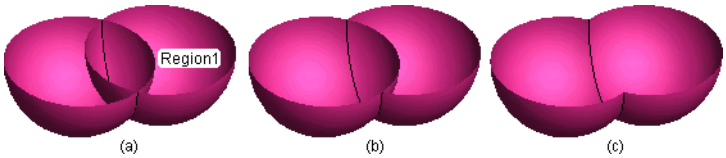


Figure 4-16: Illustrations of the simplify operation

4.7.10 Copy objects

Geometry objects, including parents of objects as shown in the tree, can be copied to new root-level parts by selecting the items and selecting *Edit* → *Copy* from the main menu (or using the short-cut <Ctrl><K>). For example, if a spherical shell is created by subtracting one sphere from another (which will remove both original spheres from the model and list them as parents of the Boolean operation) the inner sphere can be copied to create a new root-level part which fits inside the shell. Copied items are created in the

root level, even if the original items form part of an assembly. It is also possible to make multiple transformed copies of parts. See section 4.7.1 above.

The new part is completely independent of the existing one and neither will change if the other is modified. If it is desired that they change together, the original object should be created using variables as these are maintained during the copy.

It is also possible to select faces or edges of parts and select *Copy*. In this case new parts are made for each selected item. This allows, for example, an edge to be copied from a complex object and used to loft to another edge. Note that these copies are snapshots of the model when the copy is made — they are not linked to the parent object and are not parametric.

4.7.11 Copy original

If faces or edges are deleted from a part, they are removed from the model. These surfaces can be recreated by copying the parent objects of the affected part and redoing the operation which created the part. This can be a very cumbersome operation for which CADFEKO provides a simple solution. Select the required items and select *Edit* → *Copy special* → *Copy original* from the main menu or *Copy special* → *Copy original* from the pop-up menu.

4.7.12 Exploding parts

Selecting the *Geometry* → *Explode* menu item will explode all selected geometry parts. Separate new parts are created for each face and free edge of the original parts. The new parts represent a snapshot of the geometry at the time it was exploded — they are not parametric.

4.8 Specifying dielectric media

Before it is possible to use dielectrics in a CADFEKO model, the dielectrics must be added to the *Media* list in the contents tree. By default this list contains only *Free space* and *Metal solid*. Dielectrics can be added by right– or double clicking the *Media* entry or by selecting *Geometry* → *Create dielectric medium* from the main menu.

The square icon next to the dielectric name in the *Media* list, shows the colour used to display this dielectric when displaying the geometry in *Colour by medium* mode (see section 4.3.4). This mode can be activated — per view — by clicking on the arrow next to the *Show / hide geometry* icon on the 3D view toolbar. The display colour for each medium can be set by right or double clicking the corresponding medium entry in the *Media* list.

Dielectric properties are specified on regions. The regions of each part are listed in the details tree under *Regions*. These are the individual volumes of the part. Sometimes imported models will use surface parts to define the boundaries of solid regions. These models then have no regions and it is not possible to specify dielectric properties. However, if surface parts are unioned to form closed surfaces, CADFEKO automatically constructs the applicable regions for the new parts.

To set the medium parameters, select one or more regions and select *Properties* from the pop-up menu. This opens the *Region properties* dialog as shown in figure 4-17. The *Mesh size* field allows defining a local mesh size (see section 4.12.1) for tetrahedral meshing.

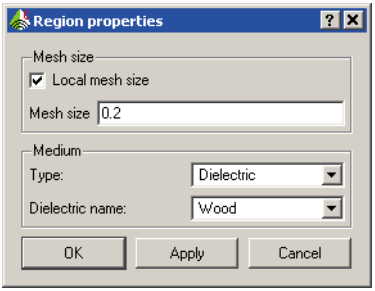


Figure 4-17: The *Region properties* dialog

The *Type* field is used to define the medium type — it may be metallic, dielectric or free space. Free space indicates the surrounding background medium which may be a homogeneous dielectric, but is usually free space. Free space regions are hollow while metallic regions are solid. The difference is mainly apparent during geometry operations.

The *Dielectric* option is only allowed if the *Media* list contains dielectrics as discussed above. (Dielectrics can be defined while the *Region properties* dialog is open — the *Dielectric* option then becomes available immediately.) For dielectrics, the *Dielectric name* must be selected from the list of available dielectrics. This name is used with the DI card in EDITFEKO to set the medium parameters. Unlike label names, which must be unique, the same medium name can be used for many different regions. For Boolean unions between bodies where intersecting regions have conflicting material properties, an *Unknown* medium is created automatically.

Initially the dialog shows the current state of the property for the selected item(s). If multiple regions are selected, fields which are not the same for all dialogs are left blank. These properties are not modified when changes are applied. This allows, for example, modifying the mesh size for a number of different dielectrics simultaneously.

4.8.1 Metal faces

If a region is dielectric, faces bordering it can be set as metallic by checking the *Metal surface* box on the *Face properties* dialog shown in figure 4-18.

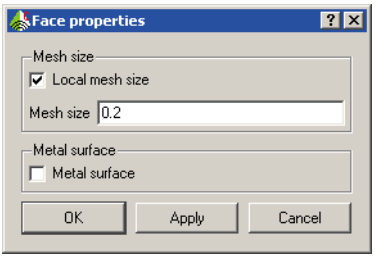


Figure 4-18: The *Mesh properties* dialog for faces

Note that faces bordering a metal region or those with free space on both sides are always metal and for these the *Metal surface* field is disabled. However, the property is persisted. Hence if a dielectric region with a metal face is set to a metal solid, all its faces become metal, but if the region is set to dielectric again, only the original metal face will again be metal. The details tree uses icons to indicate the metal properties of a face. If the face borders a solid metal region or two free space regions (the *Metal surface* property is ignored), no special icon is shown. If the property can be set, the icon indicates if it is set to metal or dielectric.

If multiple surfaces are selected, the *Metal surface* item is disabled if any of these surfaces cannot be set. If all surfaces can be set, but do not have the same state, the *Metal surface* item is in a ternary (i.e. not defined) state. This indicates that the existing states will be persisted and allows, for example, setting a local mesh size on a number of surfaces without changing their metal property. Clicking on the checkbox cycles through checked, unchecked and ternary.

The *Mesh size* field allows defining a local mesh size (see section 4.12.1).

4.9 Geometry validation

CADFEKO provides tools to perform basic consistency checks on the geometry. Since properly connected meshes are only guaranteed for single parts, it is important to ensure that parts do not clash. Parts clash if there is any contact between them or if one part is completely inside another. Select all the items to test (or press <Ctrl><A> while selecting geometry parts) and select *Find* → *Clashing geometry* from the main menu. The clashing parts remain selected and are listed in the message window. The selection undo / redo operations can, of course, be used to see what was selected before the operation.

Find → Unconnected geometry edges finds and selects all the edges of the selected parts that are free (not attached to any face) or attached to only one face. If this operation selects an edge that seems to be connected to multiple faces, it means that there is more than one edge at this location and that the faces do not make electrical contact. This usually indicates a problem in the model.

4.10 Assemblies

Assemblies are used to organise the geometry. They become part of the geometry structure and are shown as the first level under *Geometry* in the tree. To create a new assembly, select the required items, right click in the tree or 3D view and select *Assembly → Create*. Only parts (displayed objects) can be added to an assembly. (Mesh and geometry parts cannot be added to the same assembly.) Items are moved between assemblies by selecting *Assembly → Move to → ...* in the pop-up menu. Similarly *Assembly → Move out* moves the selected items back to the root level. An item cannot be in two assemblies at the same time.

Deleting an assembly removes the assembly and its contents from the model. Similarly, selecting an item in an assembly and pressing <Del> will remove it from the model. (It is very different from *Assembly → Move out*.) Selecting *Disassemble* on an assembly moves the items inside it back to the root level and deletes the assembly — in effect removing the assembly without deleting the contents.

Operations (Booleans, transforms, splitting, etc.) are applied to the individual items. If multi-parent operations are applied to items in one assembly, the result is also in that assembly. Where such operations are applied to items from more than one assembly, the result is placed at the root level.

Assembly names are part of the full label / name, for example, `Assembly1.Union4.Face12`. This is the reference required in EDITFEKO to refer to elements on this specific face. Note that all names in CADFEKO are case insensitive, i.e. `Face1` and `face1` are the same.

4.11 Selecting items

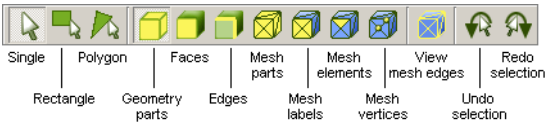
Items are selected by clicking either in the tree or in any 3D view. Selected items are highlighted in the tree and all 3D views. When parent objects are selected in the tree, CADFEKO draws wire-frame outlines of these objects in the 3D view. These items are not themselves part of the model, but it is very useful to be able to distinguish the different parents. If the current dialog operates on items that are no longer selected, these items are shown on a green background in the tree.

Pressing <Ctrl> while selecting items, in the tree or a 3D view, will add them to the current selection (or remove them if they are selected). Pressing <Shift> while clicking on items in the tree will select everything of the same type from the first item of the current selection. Since the order of items is not as apparent in the 3D view, pressing <Shift> while selecting will merely add / remove the new item. In the tree navigation can also be done with the arrow keys. This includes pressing <Shift> to select multiple items.

All selected items must be of the same type, i.e. it is impossible to select a part and some of its faces at the same time. However, simultaneously selecting faces belonging to different parts is allowed.

If the <Ctrl> key is pressed while right clicking to activate a pop-up menu, the current selection will be left unaltered. If <Shift> is pressed, the new item will be added to the selection if it is not selected, but it will not be removed if it is selected.

Selection in the 3D view is controlled with the *Selection* toolbar. This contains two sets of toggle buttons and undo and redo buttons for the selection. In each toggle set, one button is always on (active). The toolbar looks as follows



The three "how to select" buttons enable specification of how items will be selected. If the first (*Single select*) is toggled on (down), items are selected by clicking on them. If *Rectangle select* is toggled on, a rectangle is defined by clicking at one corner and then moving the mouse (with the mouse button up) to the opposite corner and clicking again. The proposed rectangle is drawn while the mouse is moved. While the rectangle is drawn, it is not possible to rotate / pan / zoom the model. All items fully or partially inside this rectangle are selected. Finally, if *Polygon select* is toggled on, a polygon is specified by clicking at successive corners. The polygon is closed by clicking on the first corner again. As an aid, the circle at this corner goes blue when the mouse is moved to within a certain tolerance of the corner.

Note that the presence of cutplanes (see section 4.3.5) has a significant impact on *Rectangle* and *Polygon select*. In the absence of cutplanes, only the visible items either fully or partially inside the rectangle / polygon are selected. Note, however, that if the *Face displacement* on the *Rendering options* dialog (see section 4.3) is too low, faces immediately behind sharp corners may be "visible". When cutplanes are present, all items are selected — visible objects as well those that are completely hidden behind other items.

The selection techniques may be mixed. For example, select all the items inside a polygon, then click the *Single select* button and add to the selection by pressing <Ctrl> or <Shift> and clicking on additional items.

The seven "what to select" buttons set part, face, edge, mesh part, mesh label, mesh element or mesh vertex selection. With selection by "mesh label", all elements with the same label are selected / unselected together. Operations are applied to the selected label. For "mesh elements", the selection and operations apply to the individual elements. For example, if a mesh label is selected and renamed, the name of this label is changed. If, however, a number of mesh elements are selected and renamed, a new label is created and all the selected elements will now have this label. Selecting items in the tree activates the appropriate "what to select" button. Pressing <Ctrl><A> will select all items of the current selection type.

Before a selection mode is changed, the existing selection is first cleared — with one exception. When mesh labels are selected and the selection mode is switched to mesh elements, the selection is converted. Hence all elements remain selected, as if they were selected as mesh elements.

If the *View mesh edges* button is toggled on (down), the edges of the selected mesh elements are highlighted. This allows finding narrow elements where the surface may not be visible on its own. While rotating the model (i.e. before releasing the mouse button), the mesh surfaces are hidden. Hence if the geometry is hidden (see section 4.3.4), rotating the model allows viewing the edges of the selected elements, even those normally hidden behind other elements.

The *Undo* and *Redo* buttons on the selection toolbar allow reverting back / forward to the previous / next selection states, respectively. These states are stored in yet another undo / redo list. This list is cleared each time the model is changed (for example, when adding / deleting / modifying geometry or variables). Note that, just like the model manipulations, selections are independent of 3D views (i.e. they are shared between them).

4.12 Creating meshes

Select *Mesh* → *Create mesh* or press <Ctrl><M> to open the *Create mesh* dialog shown in figure 4-19. Indicate if all parts or only the selected parts should be meshed. Unlike other dialogs, the selection can be changed while the dialog is open. This allows, for example, meshing different parts with different mesh sizes without closing the dialog.

The *Edge length* field specifies the default mesh size for the edges of triangles and tetrahedra. Note that some edges may be as much as 30% larger than this. (See section 4.13.1.) All free edges (edges not forming the boundary of a surface) are assumed to be conducting wires and meshed into segments. The *Segment length* field specifies the maximum length for these segments. Normally, the radii of these wires are specified on the geometry edges — see section 4.12.1 below. The *Wire segment radius* field specifies the radius which is applied to all free edges for which no local radius is specified. If *Enable volume meshing* is checked, all dielectric solids are meshed with tetrahedra. See section 2.2.2 for a discussion of the required mesh sizes.

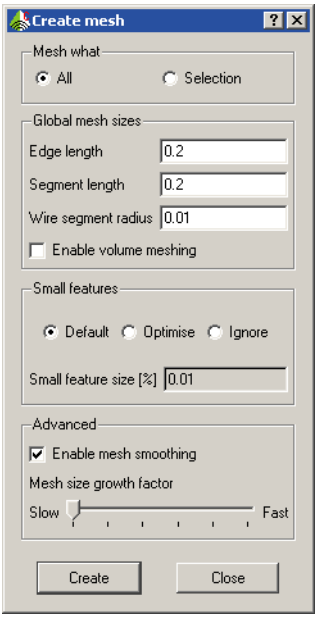


Figure 4-19: The *Create mesh* dialog

The mesh size is not allowed to be smaller than the maximum coordinate divided by 1×10^8. This is the limit of the numerical precision of the geometry. For very small models (requiring such small mesh sizes), the geometry extents (see section 4.6.1) should be decreased. The mesh size is also not allowed to be larger than four times the extents.

The *Small features* group allows special treatment of small geometry details. The *Small feature size* field specifies the limit of what is considered a small feature as a percentage of the part it belongs to. The *Default* behaviour is to mesh these structures normally. This will result in an accurate representation of the geometry, potentially using very small elements. *Optimise* is useful where the geometry has long narrow slivers or faces that are close together. If this option is selected, CADFEKO will try to align vertices on opposite sides of the small feature as shown in figure 4-20. Finally, it is possible to *Ignore* small features. Small details are then ignored at a possible cost of accurate geometry representation. This option also sometimes allows meshing faulty surfaces that cannot be meshed with the default settings. Note that ignoring small features does not work for closed edges. Such edges can, however, be broken in two by imprinting points (see section 4.7.7).

If *Enable mesh smoothing* is checked, an additional smoothing algorithm is applied. This will result in a better quality mesh, but will obviously increase meshing time.

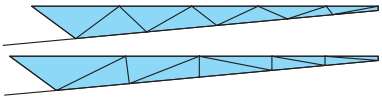


Figure 4-20: Default (top) and optimised (bottom) mesh near the tip of a long narrow element. Here the width of the strip falls within the small feature size. For the optimised mesh, the vertices on the long edges are aligned.

The sizes specified on the *Create mesh* dialog are used on all items which do not have a finer local mesh size — see section 4.12.1 below. Note that specifying a local mesh size on a region will also set that size on its bordering faces. (This only applies when the region mesh size is used, i.e. when the region is meshed into tetrahedra.) An additional finer mesh size can, of course, be specified on these faces. (The minimum of all applicable mesh sizes is used, thus local sizes that are larger than the global size are ignored.) Likewise, setting a mesh size on a face also sets that size on its bounding edges. If a finer mesh size is specified on, for example, an edge of a surface, then the triangles of the surface honour this length along the edge, even though the rest of the surface may have a much larger mesh size. The *Mesh size growth factor* controls how quickly the mesh size changes. *Fast* allows an abrupt jump from small to large elements, while for *Slow* each triangle will not be more than twice the size of the one it is connected to.

The mesh is a snapshot of the current geometry. Any changes to the geometry (for example, changing a wire radius) made after meshing will not be reflected in the mesh.

The mesh is listed in the tree under *Meshes* with the same structure as the original geometry parts. The meshes are identified by their labels only. When meshing the geometry `Cuboid1`, CADFEKO will give a warning if a mesh with such a label already exists. The existing mesh may be kept or replaced. If the mesh or geometry is renamed before trying to remesh the geometry, an additional mesh part is created. (See also section 4.13.2.)

4.12.1 Specifying mesh parameters

When regions, faces or edges are selected, local mesh parameters can be specified by selecting *Properties* from the pop-up menu. In all three cases the properties dialog includes a *Mesh size* group as illustrated at the top of the *Edge properties* dialog in figure 4-21. For regions, the local mesh size is only used when meshing the region into tetrahedra. The final mesh size on any item is the minimum of all sizes applicable to it. This includes the global mesh size. In an extreme example if an edge borders a surface which is the boundary of a region and all three items have a local mesh size, the final mesh size on the edge is the minimum of all three these sizes and the global mesh size.

For faces and regions the properties dialogs also allow setting material properties (see section 4.8).

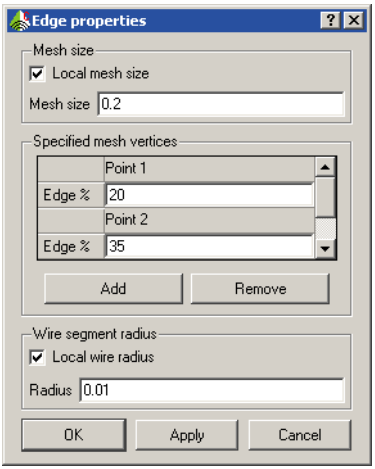


Figure 4-21: The meshing properties for edges

Initially the dialog shows the current state of the property for the selected item(s). If some selected regions have a local mesh size and others not, or if the specified size is different, the *Local mesh size* item is in a ternary (i.e. not defined) state. If this is the case, the parameters are not updated when clicking *OK* or *Apply*. This box can still be checked or unchecked to set the properties on all selected items.

The *Edge properties* dialog also allows specifying fixed mesh vertices. These are expressed as a percentage of the length along the edge. These points can be specified with the mouse (see section 4.6.3) in which case CADFEKO calculates the percentage for the point on the edge which is nearest the specified point. The percentage is then stored with the edge, i.e. if the object is later scaled or moved the specified vertex will still lie at that percentage from the edge. Vertices can also be specified using named points. In this case, the point is stored with the edge, and if the object is later moved or scaled, the specified vertex may no longer be on the edge.

For free edges (i.e. edges not forming the boundary of a surface), a wire radius can be specified on the *Edge properties* dialog. Note that wires are displayed as lines in the 3D view, i.e. the radius is not represented.

4.12.2 Importing meshes

CADFEKO can import a variety of mesh formats, accessed with *File* → *Import* → *Mesh* → ... in the main menu.

For *FEK files* only the geometry section (wire segments, triangles, unmeshed polygons and tetrahedra) is imported; the control section is completely ignored. Since CADFEKO

specifies dielectric medium boundaries based on face labels, it appends an underscore and a number to create unique labels if the FEK file contains different dielectric boundaries with the same label. In addition, since the imported labels are listed under a new mesh part, the labels in the imported model contain an additional level. The medium information and segment radii are retained.

CADFEKO imports the other mesh formats by running PREFEKO and importing the resulting FEK file. Currently FEMAP neutral files (boundary surfaces bordered with line curves are imported as mesh polygons), NASTRAN, PATRAN, STL, meshed AutoCAD DXF files (only the LINE and POLYLINE structures which defines segments and triangles), Ansys CBD and Concept files are supported. For information on these formats, see section 9.2.18.

For all formats except FEK files, CADFEKO opens an additional *Import mesh* dialog. Here the user can set a scale factor and control which elements to import. (For example, boundary elements in FEMAP neutral files may be either part of the geometry — i.e. it should not be imported — or intended as mesh polygons — i.e. it must be imported.) *Quadrangles* are divided into triangles during the import process. Only Ansys CBD files support segment radius information — for all other formats a default radius must be specified on the *Import mesh* dialog. (The default radius is also used for Ansys segments where a radius is not defined in the file.) Different radii can be set on the segments after importing. Finally, for AutoCAD DXF files, the LINE elements are divided in segments according to the value of the *Segment length* field. If the LINE elements must not be subdivided, this value must be larger than the longest line. Since these formats do not support specifying dielectric media, all segments, triangles and polygons are in free space and all tetrahedra have the medium *Unknown*.

It is, of course, also possible to import meshes manually. The user constructs a simple PRE file that imports the mesh in EDITFEKO, run PREFEKO and then import the FEK file in CADFEKO. This allows more control over the import process, for example to only include certain labels.

4.13 Mesh validation and editing

4.13.1 Mesh information

Selecting *Mesh* → *Info* from the main menu shows a histogram of the edge length distribution and the number of the different elements in the current selection. (The selection may contain mesh parts, labels or elements.) The spread gives an indication of the quality of the mesh and shows how many edges are over a desired length. The numbers of the different elements indicate the problem size and the required run time.

4.13.2 Relabelling mesh elements

A selection of mesh elements can be relabelled provided they are of the same type and belong to the same mesh part. Since the full label is constructed as `Assembly.Part.Label`, it is not possible to relabel elements on different parts. Note that the individual elements, not the labels, must be selected (see section 4.11).

Labels are used to reference specific elements when setting up the control parameters in EDITFEKO. To specify unique properties on an element or collection of elements, those elements need to be relabelled. As an example, consider a simple wire monopole attached to a plate, created as `Union1`. The resulting mesh contains two labels: `Face1` for the triangles on the plate and `Wire5` for the segments on the monopole. The bottom segment can now be selected and its label changed to `Feed`. In EDITFEKO this segment can now be referenced by its full label `Union1.Feed` in any field expecting labels. Relabelling may also be used to reduce the number of labels in any given part.

When the last element of any label is removed (deleted or relabelled), the label is removed.

Care must be taken when remeshing parts where the mesh has been relabelled. Since the entire mesh part is replaced by the new mesh, the label created by relabelling the elements is lost. If the example above is remeshed, there will again be only the labels `Face1` and `Wire5`. Therefore, relabelling elements should be delayed as much as possible. (Another option is to break the monopole in two in such a way that the bottom wire will always mesh into only one segment, but then the geometry must be modified before generating a new mesh with a different size.)

4.13.3 Transforming mesh parts

Mesh parts can be transformed similarly to geometry parts (see section 4.7.1). The transform applies to the entire part and can be edited or removed. If a vertex of a transformed mesh is edited, the transformed coordinates are displayed and user input is assumed to be in the transformed space. If the transform is modified, the effect is also applied to the edited vertex.

4.13.4 Merging coincident vertices

Connectivity in FEKO relies on vertices of adjacent mesh elements being within a small tolerance of each other. CADFEKO allows merging all vertices within a user-specified tolerance. To do this, select one or more mesh parts and select *Mesh* → *Merge coincident vertices* from the main menu to open the *Merge coincident vertices* dialog. Here a *Tolerance* can be specified. Any two points separated by less than this distance are then merged to the coordinates of one of the original vertices (as opposed to taking the average position).

If *Snap to geometry points* or *Snap to named points* is on, mesh vertices lying within the specified tolerance of these points will be snapped to them. For example, if a named point lies between two mesh vertices that are less than the specified tolerance from each other they will be merged at this point.

Merging points can lead to degenerate (collapsed) triangles. CADFEKO tries to avoid this by giving a warning if the tolerance is large relative to the mesh sizes. It also automatically removes all degenerate elements after merging the vertices.

4.13.5 Merging elements (removing vertices)

The automatic mesh removal is meant primarily to merge vertices that should be the same, but differ due to numeric tolerances. If the points are further apart, it may be required to control which point is moved and which point remains fixed. It is also often required to remove specific elements where the distance between points may be large compared to the triangle size elsewhere in the model.

This is done by deleting vertices, which is a two-step process similar to the subtract operation. First select all the vertices which must be moved and press <Del>. In the second step, a final vertex is selected when prompted. All the originally selected vertices are then replaced with the final vertex. CADFEKO automatically removes all elements that become collapsed (degenerate) during this operation.

4.13.6 Remove duplicate and collapsed elements

Duplicate elements within a mesh part can be deleted automatically. (Generally duplicate elements should only occur in imported meshes or where CADFEKO meshes were edited manually.) Select the required mesh parts and select *Mesh* → *Remove duplicate elements* from the main menu.

If duplicate elements have the same label, CADFEKO deletes all but one. If the elements do not have the same label, it may be important to delete a specific element. This is determined with the *Remove duplicate mesh elements* dialog shown in figure 4-22. The items are ordered by pressing <Ctrl> and clicking and dragging the numbers in the left-hand column. For each set of duplicates an element whose label is highest on this list is retained and all others are deleted. (There could, of course, be more than two identical elements.)

Selecting *Mesh* → *Remove collapsed elements* from the main menu will delete all degenerate elements (elements where two or more nodes coincide).

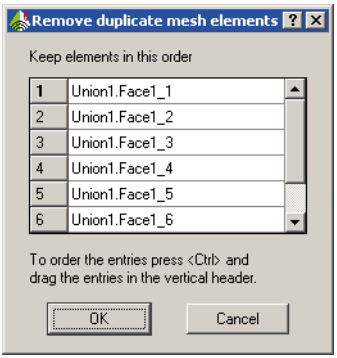


Figure 4-22: The remove duplicate dialog which determines which of the duplicate elements to retain

4.13.7 Finding distorted and oversized elements

Since the accuracy of the FEKO results depends on the quality of the mesh, CADFEKO allows some mesh validation. Selecting *Find → Distorted mesh elements* or *Find → Oversized mesh elements* from the main menu checks the current selection (mesh parts, labels or elements) for distorted or oversized elements, respectively. The result is a selection of the inappropriate elements. The selection *Undo* and *Redo* operations can be used to see which elements were selected before and after the test.

Distorted mesh elements are specified in terms of the minimum internal angle. (All angles in CADFEKO — except the arguments of trigonometric functions used in expressions — are in degrees, and this is no exception.) In an ideal mesh all internal angles are 60 degrees, and if any of the three angles are much smaller than this, the element is a sliver. These elements can be removed by deleting vertices (see section 4.13.5).

Oversized elements are found based on the maximum edge length.

4.13.8 View free mesh edges and segment nodes

Another very important issue is connectivity. In addition to the tests on the geometry, CADFEKO allows finding unconnected edges and segment vertices. This is done by selecting *Find → Free mesh triangle edges* or *Find → Free segment nodes* from the main menu. These operations test only the selected elements, and will not report free edges / vertices if the element is connected to another unselected element.

For segments, all vertices which are not attached to another segment or a triangle corner are selected. These vertices can be completely acceptable, for example, at the end points of wire antennas. However, if two such points lie very close together, they should be merged.

Triangle edges are considered to be free if they are not attached to other triangles. (An edge is only connected to a triangle if both end points coincide with corner points of the triangle.) Here the two vertices at the ends of the edge are selected. These vertices may then be merged, moved or whatever is required to make the model consistent. Again, free edges may be perfectly normal such as on the edge of a finite boundary.

4.13.9 Editing mesh vertices

A mesh vertex can be modified by selecting it (mesh vertices must be visible and mesh vertex selection on) and selecting *Edit* → *Properties* from the main menu (or *Properties* from the pop-up menu). This opens the *Modify vertex* dialog which contains the coordinates of the vertex (in global coordinates). The input fields accept point entry (see section 4.6.3) and expressions, but these are converted to their values once the dialog is closed. The point can be moved by a given amount (rather than a specific position) by adding the required offset to each field. For example, changing the X coordinate from `1.2` to `1.2+0.5` moves the point 0.5 units in the X direction.

4.13.10 Creating mesh triangles

Sometimes manual mesh fixing is required. This may be because the mesh contains a number of holes or because some bad elements were selected and deleted. (Unlike deleting vertices (see section 4.13.5), deleting elements leaves a hole in the mesh surface.)

To this end, mesh triangles can be manually constructed by selecting *Mesh* → *Create new triangle* from the main menu. This operation is only allowed if the selection contains a single mesh label (in which case the new element is added to this label), or a single mesh part (in which case the new element is added to a new label created under this part). The *Create mesh triangle* dialog allows specifying the three corners of the triangle. Since these three fields are standard point entry (see section 4.6.3) fields, all the different snap options (see section 4.3.3) are available.

The newly created triangles can be "moved" into an existing label. Select all the elements of the existing label as well as all the new elements. (If the elements are selected by label, click the *Select mesh element* button to convert the selection to individual elements.) Now right click on the selected elements and select *Rename*. This will merge all the selected elements to one label. All the labels that become empty are automatically removed and these names become available for the new label.

4.13.11 Reversing normals

Some features in FEKO make use of the normal vectors of elements. For example, dielectrics are specified by giving the medium on each side of the bordering triangles. For

this purpose, the normal side of a triangle or polygon is defined as the side towards which the normal vector is pointing. (If the start point of the normal vector is located on the triangle, the normal side is the one closest to the arrowhead.) In CADFEKO the normal sides of mesh elements are coloured blue, while the backs are brown.

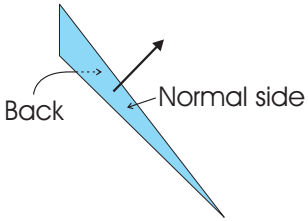


Figure 4-23: The normal and back sides of a triangle

When the selection contains triangles or mesh polygons (or labels or parts that contain such elements), selecting *Mesh* → *Reverse normals* will invert the normals of all the selected triangles / polygons. Normal vectors can be shown explicitly per 3D view by activating the *Show / hide normals* button on the 3D view toolbar. Normals are then shown as lines with a red tip.

Reversing normals of triangles on dielectric boundaries can adversely affect the consistency of the model. CADFEKO tries to keep the model consistent. For example, if the normals of all triangles in a given label are reversed, CADFEKO swaps the dielectric media on the two sides. However, if only some triangles are modified, CADFEKO cannot ensure consistency and it is up to the user to ensure that the model is consistent.

4.13.12 Setting mesh properties

Ideally properties should be set on the geometry as this is copied to the mesh each time the geometry is meshed. However, it is possible to set properties on the mesh. Select *Edit* → *Properties* to open the appropriate *Mesh properties* dialog set. Note that medium properties can only be set if the selection contains mesh labels rather than elements.

For tetrahedra only the medium can be specified. For triangles, the medium is specified on each side. If either medium is a dielectric and the other is not solid metal, then the triangles defining the border can be specified as metallic or not. For segment labels both the segment radius and the surrounding medium can be specified, but when the selection contains individual segments, only the radius can be specified. Specifying the media on the mesh elements should only be required for imported meshes.

If multiple elements / labels with different properties are selected, some input fields may be blank. If these are left blank, CADFEKO will not change those properties when other changes to the dialog are applied.

It should be noted that these properties are associated only with the mesh. If the model is remeshed, they are lost (unless the mesh was renamed first, in which case there will be a large number of duplicate or clashing elements). Hence it is not recommended to set mesh properties on meshes where the geometry is available. These tools should be reserved for use on imported meshes.

4.14 Preferences

Select *Options* → *Preferences* to set the default coordinate system for geometry primitives (see section 4.6), and the number of undo steps to keep in memory. Since a large amount of memory is used to make provision for undo operations on the mesh, reducing the number of undo entries can significantly reduce memory usage during mesh manipulation.

4.15 Running components

The CADFEKO model is saved as a CFX file. Solving this model with FEKO requires two additional input files, namely the CFM file which contains the mesh information and the PRE file which controls the solution process. CADFEKO automatically exports the current mesh to the CFM file each time a component is started from the *Run* menu. The CFM file can also be exported directly (see section 4.2.2) using *File* → *Export* → *CFM file*. This allows, for example, updating of the model and then running PREFEKO from EDITFEKO if the latter is already open. CADFEKO also creates a default PRE file — which imports the CFM file — the first time the model is saved. Since this file is edited (by EDITFEKO) to specify the control parameters, it is never overwritten by CADFEKO.

4.15.1 Solution options

The *Solution options* item under the *Run* menu allows specifying parameters for the various FEKO components. (See See Chapters 7 and 8.) The *Advanced* fields on these dialogs allows manually typing the options as would be done after the filename in a command line.

4.16 Working with CADFEKO models in EDITFEKO

Currently all control parameters (frequency, excitation, etc.) must be specified in EDITFEKO which can be launched from the *Run* menu in CADFEKO.

4.16.1 Setting units

CADFEKO has no direct support for specifying units. FEKO, however, expects all input to be in metres. The SF card in the geometry section of the PRE file can be used to specify the units. For example, if the model was constructed in millimetres, an SF card with a 0.001 scale factor should be added to the PRE file.

4.16.2 Referencing elements

In EDITFEKO, properties can be set on specific elements using their labels (see section 4.10). Segments have the label of the edge (typically called `Wire...`), triangles that of the face (typically `Face...`), and tetrahedra that of the dielectric region (typically `Region...`). Of course, these names can be modified on the geometry or the mesh elements. See section 4.13.2 for more details.

Since every region / face / edge has a unique label, a typical model may contain a large number of labels. Setting properties such as losses then requires a large number of cards. To simplify this, PREFEKO supports renaming multiple labels using the wild-cards `*` and `?` (where `*` expands to any string and `?` to any character). For example, renaming `Wheel?.Face*` to `Wheel` will rename the faces `Wheel1.Face1` and `Wheel2.Face17` (and all others matching this pattern, but not `Wheel10.Face1` or `Wheel1.Wire1`) to `Wheel`. They can then be referenced with a single name. After renaming these elements, the original names no longer exist, i.e. referencing items using those names will not find any elements.

4.16.3 Using variables and named points in EDITFEKO

When exporting the CFM file, CADFEKO includes all variables and named points with their current values. These values are then imported by PREFEKO and can be used in the PRE file — at any point after the IN card.

4.16.4 Setting dielectric parameters

In CADFEKO, dielectrics are defined on the regions of complex bodies (see section 4.8). Imported models often only have faces. For this reason, CADFEKO automatically constructs regions as part of a union of faces that form a closed body.

For all dielectrics, the medium name is specified in CADFEKO and used (in the DI card) in EDITFEKO to specify the material properties for this dielectric. The dielectric names are not hierarchical like the label names. This allows using the same name on completely separate parts.

4.16.5 Using symmetry

Currently symmetry is specified with the SY card in EDITFEKO. This generates new elements, i.e. only one half of the model should be built in CADFEKO. Typically, the completed model is created and then split along one or more of the principal planes. Structures on the symmetry plane require special care.

When splitting a solid, a new face is created along the split plane. If the entire model is meshed, this face will be included. These elements will then be duplicated by the symmetry operation, which will result in an invalid mesh. The simplest solution is to convert the solid to a shell before splitting it, but this option may not always be possible (for example, if the solid is a dielectric). Alternatively these elements can be deleted after meshing, or the IN card adapted not to include them.

A more complex problem involves models which contain structures on the symmetry plane. These have to be imported after specifying the symmetry plane. First the IN cards with label selected importing should be used to import all geometry not on the symmetry plane, then the SY card is specified and finally the structures on the symmetry plane are imported.

FEKO supports ranges of labels if the strings end in numbers. So only one card is need to include, for example, all the labels from `Assembly.Part.Face3` to `Assembly.Part.Face12`. Ranges do not include items with completely different names, names that do not have a number or names where the number lies outside the range. Note that the number as a whole is used, not individual digits. The example will include `Assembly.Part.Face4` and `Assembly.Part.Face11`, but not `Assembly.Part.Face` and `Assembly.Part.Face44`.

Fewer IN cards are required if elements are relabelled in CADFEKO. Elements belonging to different parts cannot belong to the same label, so forming a union of the model before meshing it can simplify matters.

See the *Getting started manual* for more details.

4.17 Notes editor

CADFEKO supports a rich text editor — shown in figure 4-24 — allowing adding comments to a model. This is opened from the main toolbar or by selecting *View* → *Notes*.

The toolbar at the top contains buttons to *Clear* and *Print* the notes, *Undo* / *Redo* changes as well as the standard *Cut*, *Copy* and *Paste* commands. Multiple undo / redo commands are supported. There is no facility to *Cancel* any changes to the text as this may be achieved with the *Undo* command.

In addition, the editor allows setting the font name, font size, font type (bold, italic, underlined) and text colour. Finally it allows setting the justification. If no notes have been previously saved with the model, then a basic template is loaded.

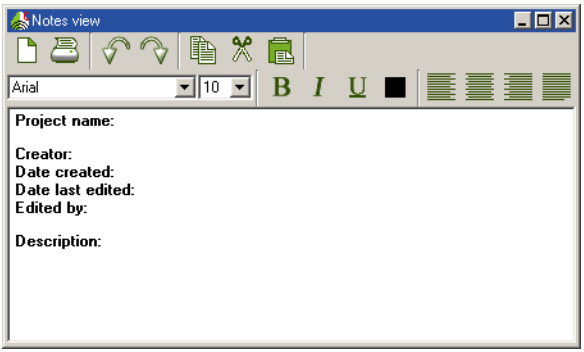


Figure 4-24: The notes editor

4.18 Getting help

Select *Help* → *CADFEKO manual* to open the online documentation in the *Help assistant*. More information on how to use the *Help assistant* can be found from the *Help assistant manual* entry on the *Contents* tab of the assistant itself. The *Help assistant* is automatically closed when CADFEKO is closed. In addition, pressing <F1> in any dialog will open online help associated with that dialog.

The *Help* menu also gives access to the *FEKO user manual* and the *Getting started manual*. These are PDF documents and the appropriate PDF viewer must be configured. Select *Options* → *Preferences* from the main menu and enter (or browse for) the path and name of a *PDF viewer* such as Acrobat reader.

Select *Help* → *About* to show the version of CADFEKO.

4.19 Short-cut keys

The following keyboard short-cut keys are available in CADFEKO

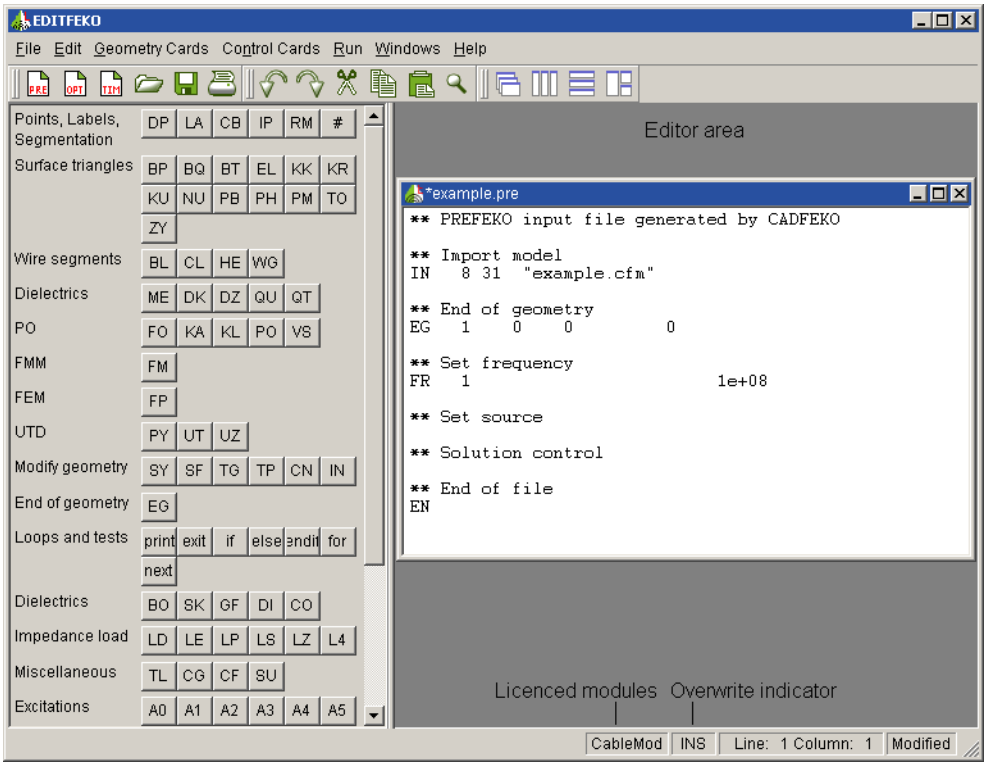
<Shift><F1>	"What's this?" (extended tool tip) help for the component that has focus.
<F1>	Context-sensitive help for the dialog / window that has focus.
<F2>	Rename selected item.
<F9>	Edit the workplane.
<Del>	Delete selected items.
<Shift><Ins>	Paste clipboard text.
<Ctrl><Ins>	Copy selected text.
<Shift><Del>	Cut selected text.

<Ctrl><A>	Select all items of the current selection type (object, face, edge, mesh element, etc.)
<Ctrl><C>	Copy selected text.
<Ctrl><K>	Create new root-level parts from copies of all selected items.
<Ctrl><M>	Create mesh.
<Ctrl><N>	Create new model.
<Ctrl><3>	Create new 3D view.
<Ctrl><O>	Open model.
<Ctrl><Q>	Select snap option (only if a 3D view has focus).
<Ctrl><S>	Save model.
<Ctrl><V>	Paste.
<Ctrl><X>	Cut selected text (only in the notes editor).
<Ctrl><Y>	Redo model creation / modification.
<Ctrl><Z>	Undo model creation / modification.
<Alt><←>	Undo view manipulation.
<Alt><→>	Redo view manipulation.

5 The editor EDITFEKO

A PREFEKO input file is a standard ASCII text file that may be created with any available text editor. The model geometry and desired calculations are entered through lines of text, referred to as cards. (The complete geometry can be build in CADFEKO and imported using one of these cards.) Each card may have a number of parameters which must appear in fixed positions. The editor EDITFEKO was designed to simplify the process of generating PRE files. It is a basic text editor with customised functionality.

The interface consists of an editor area where more than one file may be opened — each in its own edit window. The toolbars at the top provides quick access to some standard functions. The status bar on the bottom of the window contains license information, indicates when "Overwrite" is on, the line and column numbers, and also gives an indication of the file modification status. In the standard PREFEKO mode the button panel provides quick access to the card editor dialogs.



The main functionality is the specialised card editor dialogs (see section 5.4) for PREFEKO input cards. It also supports OPTFEKO input files (see section 5.5), but currently uses a plain text editor for TIMEFEKO input files.

The program may be started from a command prompt by entering

```
editfeko example.pre
```

where the optional parameter (here `example.pre`) specifies the file to open. If the parameter does not contain an extension, the extension `pre` is added. Also, if such a file does not exist, it will be created. EDITFEKO can also be opened from inside CADFEKO and POSTFEKO.

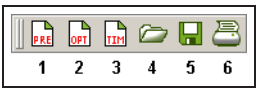
If a file is opened in EDITFEKO a lock-file is created (of the form `.filename.ext.lock`) in the same directory as the file. If you open the file in another instance of EDITFEKO at the same time, it will give a warning ("A lock file already exists for this file. ... Are you sure you want to continue with this operation?") with options to cancel or continue. Generally one should then click *Cancel* and switch to the other instance of EDITFEKO. One should only continue here if the lock file resulted after EDITFEKO was not closed properly (such as a power failure or system crash).

5.1 Toolbars

The toolbars in EDITFEKO are usually docked at the top of the EDITFEKO window, but they can be dragged free or docked on any of the four sides. They can also be switched off by selecting *File* → *Preferences* from the main menu and selecting the *Appearance* tab.

Note that the splitter between the editor and the dialog panel may be dragged to resize the panel.

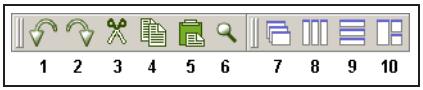
5.1.1 File control toolbar



Used for file and print control. The functions of the buttons are:

1. Create a new `*.pre` file and switch to PREFEKO mode.
2. Create a new `*.opt` file and switch to OPTFEKO mode.
3. Create a new `*.tim` file.
4. Open a file (and switch to the appropriate mode).
5. Save the current file.
6. Print current file.

5.1.2 The edit and window toolbars



The edit toolbar provides standard edit commands while the window toolbar allows positioning the edit windows in the editor area. The functions of the buttons are:

1. Undo. Multiple undo commands are supported.
2. Redo. When new text is added (by typing or adding a card) the redo chain is cleared and the new text forms the last entry in the undo chain.
3. Cut.
4. Copy.
5. Paste.
6. Search for text.
7. Cascade all edit windows.
8. Tile windows vertically.
9. Tile windows horizontally.
10. Arrange windows as best as possible.

5.2 Menu items in EDITFEKO

5.2.1 File menu

The *File* menu allows opening and closing files and is similar to standard Windows applications. Most of the options and the list of recently used files do not need any explanation.

The *New* sub menu item opens a sub menu where the user may choose **.pre* for PREFEKO, **.opt* for OPTFEKO, or **.tim* for TIMEFEKO files. EDITFEKO will enter the appropriate mode depending on the selected option. The *Save* menu will ask for a filename if the file does not have one, otherwise it will save (without confirmation) using the current filename. Various "hot-keys" are presented next to the respective menu items.

The file menu is also used to set user preferences (see section 5.3).

5.2.2 Edit menu

This menu provides general editing commands. Note that multiple level undo/redo is allowed. Each change to the editor is recorded in the undo list. When the user clicks

Undo the entry is moved to the redo list and the change to the editor is undone. When new text is added (by typing or adding a card) after a sequence of undo commands, the redo chain is cleared and the new text is recorded at the end of the undo list at that time.

It is possible to search the editor for specific text, case sensitive or not. The search always starts from the current cursor position. If it is desired to start the search from the top of the file, the cursor must be moved there (using <Ctrl><Home>). Note that there is also a fast card search. By right clicking on any button on the button panel, the next occurrence (starting from the current cursor position) of the associated card is found.

Note the *Goto line* option which is very useful to find lines reported by PREFEKO. (PREFEKO always reports the line number where errors occur.)

The *Comment* and *Uncomment* item allow the user to comment out blocks of text or remove the comments. This is often useful when debugging large files. The comments are inserted as ** followed by a space and only removed if this sequence is present.

For example, in the lines

```
** ** Select the label
** LA   1
*** Next comment
```

the comment characters will not be removed from the third line using this function. It was also not generated by the *Comment* function. These commands are also available using the "hot-keys" <Alt><C> (for *Comment*) and <Alt><U> (for *Uncomment*) or by right clicking in the editor and selecting from the resulting pop-up menu.

5.2.3 Geometry cards and Control cards menus

These two menus provide a way of opening the card editor dialog panels of the various cards when working in PREFEKO mode. Note that the cards are grouped by functionality. The functionality is also available by clicking on the appropriate buttons in the button panel.

5.2.4 Run menu

This menu is used to launch the various FEKO components. If another GUI component is started from the *Run* menu, it automatically opens the same model. Note, however, that CADFEKO does not automatically create a new model if a PRE file with that name already exists. Hence if one runs CADFEKO from EDITFEKO on a model for which only a PRE file exists, CADFEKO will open with a new nameless model. If this model is saved to the same name, it will overwrite the existing PRE file. Hence one should generally create models in CADFEKO and then run EDITFEKO. Note that when running PREFEKO or FEKO, the current file is *saved without confirmation*.

Select *Run* → *Solution options* to set the command line options for the various kernel components. The meaning of the options can be found in the chapters dealing with the separate components. The advanced fields on these dialogs allows typing parameters as one would on the command line.

It is possible to launch the solution on a remote machine. If the *Remote FEKO execution* item is checked, EDITFEKO launches the solution on the host specified in the *Remote execution* group of the *FEKO* tab of the *Solution options* dialog. Remote launching must be set up as described in the *Installation* section of the *Getting started manual.* (See section 8.2.3.)

If *Parallel FEKO execution* is checked, the solution is launched on a parallel cluster as defined in the *Parallel execution* group of the *FEKO* tab of the *Solution options* dialog. Unless the parallel job is also a remote one, the local machine must be the first host name in the configuration. (See section 8.2.2.)

5.2.5 Window menu

This menu presents the standard Windows functionality to arrange or select the editor windows on the main editor area.

5.2.6 Help menu

In addition to providing access to the version of EDITFEKO (*About*), this menu contains shortcuts to the *User's manual* and *Getting started manual*. These manuals are in PDF format and a PDF viewer must be configured on the *Preferences* dialog (see section 5.3).

The *Check for updates* item opens the *Check for updates* dialog. This polls the EMSS web-site and informs the user if any updates has been added since the last check. (Since EDITFEKO does not know if the user also have licences for other platforms, it gives an update message regardless of which platform the update is intended for.) The update checker downloads one file containing the list of updates. It does not automatically download anything else and does not send any information to the web-site. If *Activate automatic update checking* is checked, EDITFEKO polls the web-site each time it is started more than seven days after last checking the web-site. (Note that this is off by default.) It is also possible to configure a proxy server if this is required to obtain access to the internet. Click the *Check* button to poll the web-site. The *Info* tab applies to the update checker, not the available update.

5.3 Preferences in EDITFEKO

Selecting *File* → *Preferences* open the *Configure EDITFEKO* dialog. The options is then grouped under separate tabs.

5.3.1 Appearance tab

Here the user can

- Switch off the top toolbar.
- Set the printer font.
- Set the size of the screen fonts.
- Set the preferred card format.

In the column based card format, the fields are located at fixed positions and cannot be longer than 5 (string and integer fields) or 10 (real fields) characters. In the colon separated format, the different fields in each card are separated with :-characters. This format allows arbitrary long entries. If EDITFEKO is using column based format, it will automatically switch to colon separated format if any of the input fields are longer than the allowed length.

5.3.2 Options tab

Here one may

- Elect to show superuser options if this is allowed by the license. The superuser mode in FEKO is only used during program development and is not available to the general user. If one tries to edit an existing card containing superuser parameters, EDITFEKO will also prompt the user and switch to superuser mode if this is allowed. If the license does not allow superuser mode, superuser options will be removed.
- Turn on auto-save and/or backups and specify the auto-save interval.
- Specify the PDF reader required for the manuals.

5.4 PREFEKO mode

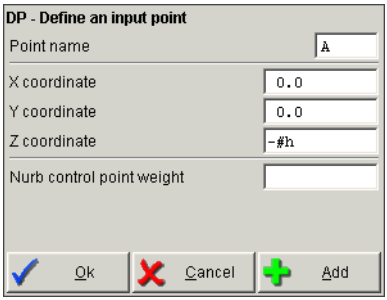
5.4.1 Generating input cards

Cards can be entered and edited directly in the editor window. In most cases the user would call one of the card editor dialogs by clicking on the appropriate button on the button panel or selecting from the *Geometry cards* and *Control cards* menus.

The buttons are grouped according to functionality namely

- Definition of node points, labels, segmentation parameters and variables (**#** button).
- Cards to generate surfaces which will be meshed into triangles elements.
- Cards generating wire segments.
- Cards for generating dielectric / magnetic volumes to be subdivided into cuboids, or specification of dielectric regions.
- Cards used in connection with the PO formulation.
- Cards to define the MLFMM and the FEM methods and parameters.
- Cards used in connection with the UTD formulation.
- Modifications to the geometry, such as translation, scaling and mirroring.
- The EG card signifying the end of the geometry. Generally the buttons above it should be used before the EG card and those below it should be after the EG card. Exceptions are the DP and IN cards and the FOR–NEXT and IF-ELSE blocks.
- Cards dealing with finite medium properties and finite grounds.
- Cards dealing with loading of wire segments.
- Non radiating networks.
- Cards to specify the excitation type, frequency and power of the excitation.
- Cards used to control which data is written to the output files.
- Cards to specify where to calculate the fields.
- The EN card which signifies the end of the input file.

A typical card editor dialog is shown below.



When a card editor dialog is selected from either the button panel or the menu, there is a button to *Add card*. This adds the card without closing this dialog. This is useful when defining a number of similar cards, for example, when specifying the first group of DP cards at the start of a model. Often only one or two parameters differ between cards. Clicking *OK* adds the card and closes the editor dialog. Pressing <Enter> has the same effect as clicking on the *OK* button. Similarly <Esc> closes the card editor dialog without adding the card as if the *Cancel* button has been clicked.

In the card editor dialogs, moving the mouse pointer over the labels may give more information of the specific input field. If the user is uncertain about the meaning or units of a given field, it is always advisable to move the mouse pointer to the label to determine if any additional information is available.

All the input fields in FEKO have a fixed length. In the card editor dialogs it is not possible to enter longer strings than allowed for the particular field. If the field contains the maximum number of characters, no new keys are excepted. Then it is necessary to delete characters before any additional ones can be entered.

Editing an existing line

The editor dialogs can also be used to edit existing lines. The line at the current cursor position may be edited either by pressing <F1> or by right clicking at any position in the editor and selecting *Edit line at cursor* from the pop-up menu. Note that right clicking does not change the cursor position. When the selection (or highlighted part) in the editor spans more than one line, the line containing the cursor is edited. For multi-line cards the cursor can be anywhere in the card.

EDITFEKO processes each card as if it does not contain errors. For some of the fields where the user may select a number of options, the editor dialog's display will default to a "common" value if the input field is invalid. *Thus it is advisable to close the card editor with* OK *even when it was used just to check the validity of the line.*

If the user types, for example, `PS` and then presses <F1>, EDITFEKO may treat it either as a new card (with default) options, or as an existing card with all options blank (zero). Setting all options to zero does not always make sense and EDITFEKO thus treats this case as a new card and uses the normal defaults. If the user wants to have a card with all options zero, he should type the card name followed by a space before pressing <F1>.

When editing existing lines, the *Add card* button is not available as the card is edited and written back to the editor.

The card editor dialogs does not edit cards generated by PREFEKO (such as the DR card). The user should rather use the input cards that are read and converted by PREFEKO. Generally, it should not be necessary to use cards that are not available in EDITFEKO.

5.4.2 Variable editor

The # button on the top right corner of the button panel launches the card editor to edit variables. This is useful as it presents a list of functions and operations understood by FEKO and calculates the value of the variable as it would be evaluated by PREFEKO at this point.

The functions, variables or operations may be selected from the three drop-down lists and a special group for the "FILEREAD" function. The selected item is automatically placed at the current cursor position. Note that variables and functions are highlighted after insertion, but operations are not. If a function is selected while some part of the variable string is selected, the selected text will be inserted as the argument of the new function. Variables, operators and the "FILEREAD" function replace the currently selected text.

5.5 OPTFEKO mode

Editing OPT files is similar to editing PRE files. All the standard editing operations behave in the same way. However, pressing <F1> anywhere in the editor brings up the OPTFEKO dialog which edits the entire file. The OPT file is divided into 4 sections, the parameter section, the penalty function section, the optimisation section, and the aim function section each with a separate tab in the dialog panel. If <F1> is pressed in any of these sections, the appropriate tab is opened. If EDITFEKO cannot determine the section, it opens the parameter tab. (The functioning of these options is discussed in Chapter 11.)

5.5.1 Parameter tab

To add an optimisation parameter, click the "insert" button to add a row into the parameter table. Similarly, to delete a row, click the "delete" button. The number of parameters specified here determines the size of the Penalty function and Gain (Optimisation tab) tables. To enforce the minimum and maximum boundaries, check the *Enforce min/max boundaries* checkbox.

The parameter tab also shows any errors in the structure of the OPT file, in the *Error* box. Double clicking on an error will highlight the error in the editor.

If the *Include default comments* is checked EDITFEKO will add the default comments before each section. The user may also type additional comments in the text editor. These comments will maintain their position also after editing the file in the dialog panel.

Note that the remaining tabs remain disabled until at least one variable is specified in the *Parameter* tab.

5.5.2 Penalty function tab

To insert a penalty function, check the checkbox. The table shows all the parameters defined in the parameter table. The user can then enter the penalty coefficients when the search moves below or above the specified range.

5.5.3 Optimisation tab

On this tab the optimisation process can be selected. Note that the size of the *Discrete points* table is also dependent on the number of variables specified on the *Parameters* tab.

5.5.4 Aim function tab

This tab allows specifying the aim or goal of the optimisation. The options for the various aim functions are explained in the OPTFEKO chapter (see Chapter 11). Multiple aim functions may be defined by clicking the *Add aim function* button (with the green plus icon). Similarly, the current aim function may be removed by clicking the *Remove aim function* button next to it. The last aim function cannot be removed.

For *Radiation pattern,* checking *Min/Max value* adds a column to the table and allows specifying a target range rather than a single target value for each point.

For *Near field values* the number of lines in the values table is the product of the *Number of blocks* and *Lines per block* fields.

5.6 Important keystrokes

The arrow keys as well as <Pg Up> and <Pg Dn> behave in the normal fashion. The following keys may be different to some other applications.

Cursor Movements

Move a word left	<Ctrl><Left Arrow>
Move a word right	<Ctrl><Right Arrow>
Move to top of visible page	<Ctrl><Page Up>
Move to bottom of visible page	<Ctrl><Page Down>
Move to beginning of line	<Home>
Move to beginning of line	<End>
Move to beginning of file	<Ctrl><Home>
Move to end of file	<Ctrl><End>

A block may be selected using the mouse, or pressing <Shift> and using the normal movement keys. If a block is selected, it will be overwritten when a key is pressed. The following list of "hot-keys" are often used

"**Hot-keys**"

Copy to clipboard	<Ctrl><C> or <Ctrl><Ins>
Cut (delete) to clipboard	<Ctrl><X> or <Shift><Del>
Paste (insert) from clipboard	<Ctrl><V> or <Shift><Ins>
Save	<Ctrl><S>
Save all files	<Ctrl><A>
Edit line	<F1>
Comment line(s)	<Alt><C>
Uncomment line(s)	<Alt><U>
Find	<Ctrl><F>
Find next	<F3>
Find and replace	<Ctrl><R>
Run PREFEKO	<Alt><2>
Run POSTFEKO	<Alt><3>
Run FEKO	<Alt><4>
Run OPTFEKO	<Alt><6>
Run TIMEFEKO	<Alt><7>

Right clicking with the mouse on a card panel button searches for the next occurrence of that card.

6 Working in POSTFEKO

6.1 Introduction

POSTFEKO is used to confirm the correctness of the model before running a potentially time consuming FEKO solution. It is also used to display the results of the FEKO simulation.

Multiple models, with their geometry (in FEK files) and results (in BOF files), can be displayed in a single POSTFEKO session. The displays are automatically updated each time a new PREFEKO / FEKO run is executed. All the display windows and their settings are saved to the POSTFEKO session (PFS) file. Note that the results are not stored in the session file. If a saved session is reopened, POSTFEKO attempts to read all the FEK and BOF files that are required to re-create the session. If a session file is copied, all the associated files must be copied *in the same path relative to the session file*. If any of these files are not available when loading a session, POSTFEKO will ask for replacement files. A POSTFEKO session can therefore be used as a template to create the same set of displays for a number of different files. POSTFEKO does not automatically save session files, and a session remains unnamed until it is saved.

The POSTFEKO rendering requirements and configuration are the same as for CADFEKO. See section 4.1.2.

6.1.1 Starting POSTFEKO

POSTFEKO can be started by selecting *Programs* → *FEKO* → *POSTFEKO* from the Windows *Start* menu or by typing `postfeko` from a command prompt. However, in most cases POSTFEKO will be started from EDITFEKO or CADFEKO. See the *Getting started manual* for an overview of the program flow.

When starting POSTFEKO from a command prompt, a filename may be specified as a command line parameter. If no extension is given, POSTFEKO will search for PFS, PRE, FEK and BOF files, in this order, with the specified name. When loading a PFS session, POSTFEKO loads all the files which were active when that session was saved. When called with a PRE file as argument, POSTFEKO will attempt to load the associated FEK and BOF files. If these files are not available at startup, POSTFEKO will load them as they become available. For FEK and BOF files it just loads the specified file. This can, for example, be used to display results without loading the FEK file if it is extremely large.

If only a FEK file is loaded, only the geometry can be displayed. POSTFEKO will, however, automatically load the required BOF file if the user chooses to display 3D data. (This is associated with the same model and hence there are no ambiguities.) To display 2D data (or plots), the BOF file must be loaded explicitly.

If POSTFEKO is started without an argument, it will load the last active session provided that session was not unnamed (and hence not saved). If started from EDITFEKO, POSTFEKO will open a new session and load the FEK and BOF files if they are available or when they become available. This makes it possible to view the geometry and plot results. It is also possible to load additional BOF files and compare results.

6.2 POSTFEKO overview

The POSTFEKO layout is similar to that of CADFEKO and EDITFEKO. At the very top is the title bar and menu. The dockable main toolbar is, by default, located just below the menu. The control panel, located on the left of the display area, controls the display of the active window. This panel contains different pages that control different aspects of the view. The different pages are accessed with a vertical row of buttons on the left side of the control panel and, where multiple pages are allowed, with the tabs at the top of the pages.

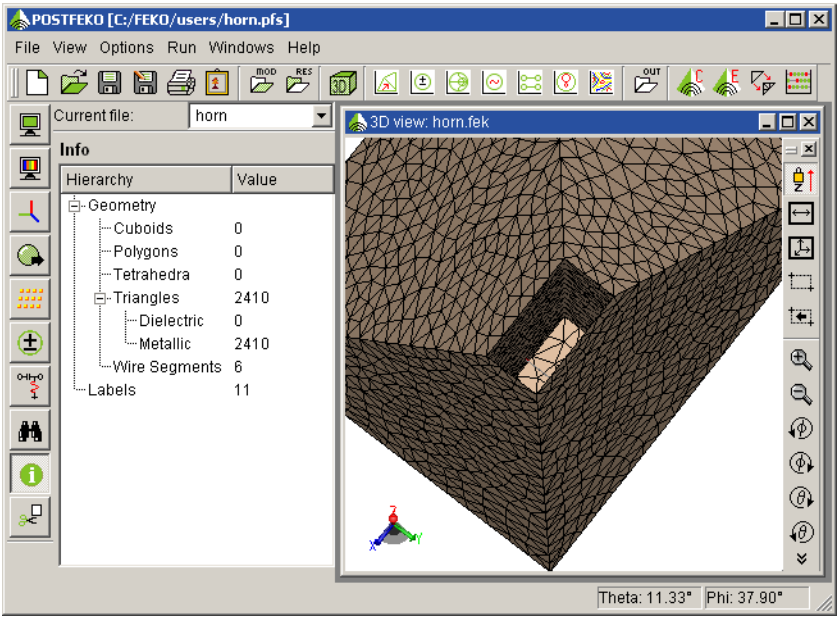
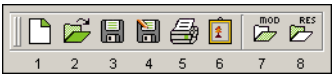


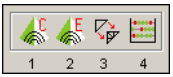
Figure 6-1: The main POSTFEKO display. Here only the geometry of a single model is displayed and the control panel is showing the model information.

The *Control toolbar* is the left part of the main toolbar and controls session information and storage. These functions are also available under the *File* menu.



1. *New session* closes the current session (and unloads all loaded model files), and opens a new blank session.
2. *Open session* closes the current session (and unloads all loaded model files), and loads an existing session.
3. *Save session* saves the current session for future use.
4. *Save session as* saves the current session with a new name or to a different directory. It only changes the session name — the names of the loaded model and result files remain unchanged.
5. *Print* prints the active display or graph.
6. *Export to image* exports the active window to an image file.
7. *Load model* loads a geometry model (FEK) file into the current session.
8. *Load results* loads a result (BOF) file into the current session.

The *Run toolbar* provides quick access to the most used components.



1. *Run CADFEKO* launches CADFEKO.
2. *Run EDITFEKO* launches EDITFEKO.
3. *Run PREFEKO* launches the model integrator PREFEKO.
4. *Run FEKO* launches the solver FEKO.

If only one model is loaded or a 3D visualisation window (which displays a single model) has focus, the component is started with the corresponding model. Otherwise POST-FEKO will prompt for a model.

For some components, optional parameters can be specified using the *Run* → *Solution options* menu item. (See also Chapters 7 and 8.) The *Advanced* fields on these dialogs allow manual entry of the options as would be done after the filename in a command line.

6.3 Displaying the geometry and model validation

A new 3D view window is created by pressing the *Add a new 3D view* button on the main toolbar. Each 3D window can only show one model (geometry and results) as shown in the *Filename* field at the top of the control panel. If the filename is very long such that it is not fully displayed in this field, hovering the mouse over it displays the full filename. Note also that it is possible to use the left and right arrows to scroll the filenames while selecting them in the drop down list. POSTFEKO automatically creates a new 3D display window each time a new FEK file is loaded with *Load model*.

6.3.1 Zooming / rotating / panning the model

The model can be rotated by clicking and dragging with the left mouse button on the 3D window. (Clicking and dragging refers to moving the mouse without releasing the mouse button.) Pressing <Ctrl> while clicking and dragging the mouse pans (moves) the model and pressing <Shift> while clicking and dragging the mouse up and down zooms in and out. Note that POSTFEKO tries to zoom towards the point where the mouse is clicked. Thus clicking at the bottom and dragging the mouse all the way up zooms in, but also causes the model to move upward. In addition, rotating the mouse wheel zooms in at the position of the mouse pointer. Clicking and dragging with the mouse wheel, pans the model. The *Render control* toolbar on the 3D view provides quick access to these features. This toolbar is usually docked at the right of the view, but can also be docked at any other side.

1. POSTFEKO supports full 3D rotation of the model as well as a more intuitive *Lock z axis* option where the z axis remains vertical on the screen.
2. *Zoom to extents* fits the model to the visible window.
3. *Isometric zoom* does a *Zoom to extents* and resets the view direction.
4. *Zoom to window* zooms to a specific region of the display. Click the mouse and keep the button down while dragging the window to zoom to.
5. *Previous view* provides a multilevel view position undo.

6. – 7. *Zoom in* and *Zoom out*

8. – 11. These options increment or decrement the view angles in spherical coordinates.

12. – 15. These options move the model in the plane of the screen.

6.3.2 The model view control panel

Initially the control panel of a new 3D view shows the *Main display options* page. (See figure 6-2.) This page controls the display of the geometry.

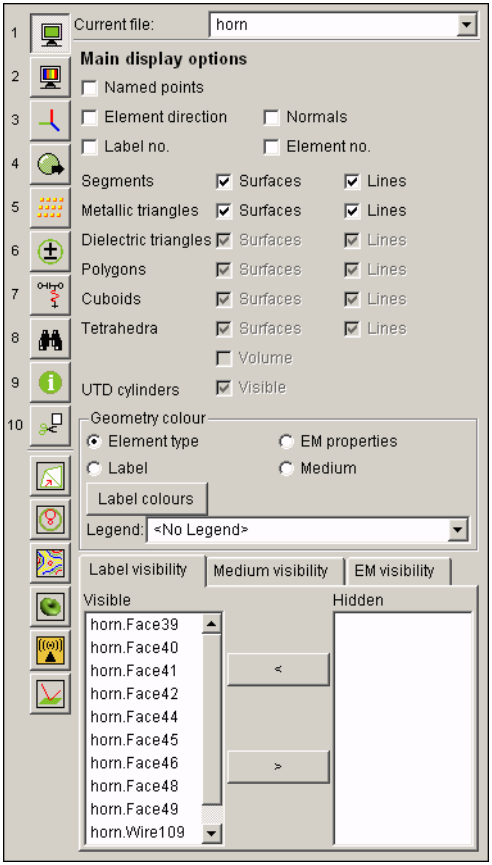


Figure 6-2: The model view control panel.

The 3D display options applies to one model at a time. This model is selected (from the list of loaded FEK files) with the *Current file* field at the top of the panel. If the model file is changed, all selected settings are applied to the new model if possible. The check boxes at the top of the page facilitates displaying specific objects and types of items. In addition, moving an item from the *Visible* list to the *Hidden* list at the the bottom of the page, hides all items with that label / medium / EM properties.

Changing the *Geometry colour* group facilitates using colour to identify labels, media, the element type (metallic / dielectric triangles, segments, tetrahedra, etc.) and the applied

EM properties (losses, PO, etc.) with the option to position the explaining legend at any of the four corners of the model. Use *Options* → *Label colours* to set the order in which colours are applied to the labels.

The different pages on the *Model view* control panel are listed below. If the *Visible* check boxes are unchecked, the associated objects are hidden or deactivated.

1. The *Main display options* page has been discussed above.

2. On the *Advanced display options* page segment radii can be enlarged to display them clearly; elements can be shrunk to better see the connecting points / edges; and the length of the normal vector can be adjusted. In addition it is possible to make the geometry partially transparent.

   This page further facilitates selecting which ground plane / Green's function card to display and how to display it. The Green's function layers can be shrinked to see the borders clearly and to view elements that lie on such boundaries. Finally, it is also possible to control the display of anisotropic dielectric layers (set with the SK card), if applicable.

3. The *Axis settings* page controls the visibility of the small orientation axis as well as the main axis. A grid can be positioned along any of the three Cartesian planes and moved orthogonal to itself. If the *Autoscale* option is off, the length of the axis can be set manually.

4. The *View transformation* page facilitates specifying the view direction, model size and light direction. The light position is defined in terms of offset angles relative to the view direction. Moving the light position can reduce bright specular reflections from small parts of the model. It is also possible to store and playback a series of view directions (camera positions).

5. The *Requested fields* page facilitates viewing the positions (specified with the FE and FF cards) where near and far fields will be calculated. The far field points are shown on a finite radius to facilitate comparison with the model orientation.

6. All the options relating to the display of excitations in the 3D view are controlled from the *Show Excitations* page. The *Set* refers to a collection of sources that are active at any given time and the *Source* field facilitates selecting individual sources from the set. Note that sources are counted with respect to how many are being displayed. If a model has four sources for some calculations, and four more sources (the first thereof a specified as a new source) are added for further calculations — the sources in the second set will be numbered 5 to 8 when "All" is selected under set, but they will be 1 to 4 when just viewing the second set. If a single source is selected, its amplitude and magnitude are displayed near the top of the page.

   For some sources it is possible to display the sources coloured according to their amplitude. For these cases, it is also possible to put a legend at any of the four corners. Four legends can be used simultaneously.

Setting the *%AI/AV/A5/A6* length controls the length of the impressed currents / dipole elements on the display. If the AI/AV sections are reduced, the it is much easier to distinguish the individual sections.

7. The *Show loading and Transmission lines* page controls the viewing of loads and transmission lines.

8. The *Find/select elements* page is used to locate elements with a specific number in the model. Ranges of values can be entered. For example, to find all triangles numbered between 3 and 5, simply enter $3-5$ in the text box next to triangles. Additional searches can be performed by comma separating values. The selection can be cleared by clicking on empty space or using the *Clear* button on this dialog.

   This page works in conjunction with selection in the 3D view. An object can be selected by clicking on it. The data associated with the object is then displayed in the view and its element number is added to the appropriate list on the find menu.

9. The *Model information* page shows information about the model. This information helps to estimate the size of the solution.

10. *Cutplanes* are used to hide the geometry on one side of the cutplane. If a cutplane is *Flipped*, the geometry on the other side of the cutplane is hidden. POSTFEKO supports multiple cutplanes. Clicking the *Add* button (the green + sign) at the top of the panel adds an additional cutplane. Clicking the *Remove* button (the green − sign) removes the current one. A cutplane can also be deactivated (without removing it completely) by unchecking the *Active* check box. The *Show all* and *Hide all* buttons activate / deactivate this *Active* check box on all cutplanes. Arbitrary planes are specified with the ϑ and φ angles of the direction normal to the plane. It is also possible to specify which types of elements to cut away.

6.4 Displaying 3D results

It is possible to view results together with the geometry in the 3D view. If the BOF file is not loaded, POSTFEKO automatically loads it if any of the 3D result display buttons are clicked. Multiple results are created by clicking the *Add* button (the green + sign) at the top of the panel. Each result can also be deactivated by unchecking *Visible* or removed from the view by clicking the *Remove* button (the green − sign) above the pages.

A legend can be displayed for any quantity with up to four legends on the display at any given time. If multiple results of the same type are displayed, separate legends can be used for each result. However, it becomes difficult to differentiate between them. It is therefore recommended to use one legend per result type and ensure that the scale limits of all the results are set to the same values.

The display is updated immediately if anything is changed on the control panel. Thus the current state of the control panel is an accurate indication of what is displayed. In

addition, all results are automatically updated each time FEKO is executed. If a results file is changed in such a way that it contains a different structure (other frequencies, additional solutions, more field calculation requests, etc.), POSTFEKO will try to maintain the selected entries, but will default to the first entry if it cannot match the existing selection. Thus the state of the selection should be carefully checked after updating to a results file with a changed structure.

6.4.1 Result view toolbar

The result buttons are used to view results in 3D. Note that clicking these buttons shows the appropriate page on the control panel and adds a result component if there is no component of the selected type.

1. The *View currents and charges* page works in conjunction with *Geometry colour* on the *Main display options* page.

2. The *View 3D far fields* page controls the display of far fields. The different polarisation options are discussed in section 6.7. The size of the pattern can be controlled with the *Scale factor* slider on the *Options* tab. The *Close* check box on the *Options* tab completes the pattern, if possible, by linking the results with the largest ϕ values back to those with the lowest values. It is also possible to move the origin of the pattern, changing the position it is displayed relative to the geometry.

3. The *View near field ortho-slices* page displays near field slices. If the *Extrusion* slider in the *Ortho slice position* group is moved to the right, larger values are drawn further from the plane, creating a kind of 3D amplitude plot.

   Under *Quantity*, with *Instantaneous* off, *Poynting* is the averaged value

   $$\vec{S} = \frac{1}{2}\mathrm{Re}(\vec{E} \times \vec{H}^*)$$

   whereas the instantaneous value is

   $$\vec{s}(t) = \vec{e}(t) \times \vec{h}(t)$$

   which could be larger than the average amplitude value.

   SAR (specific absorption ratio) is calculated from

   $$SAR = \frac{1}{2}\frac{\sigma|\vec{E}|^2}{\rho}$$

   where σ is the conductivity and ρ the medium density.

4. The *View near field iso-surfaces* page draws an iso-surface of the near field. This effectively connects all positions where the field value match the value specified in the *Iso value* slider. Note that the scale limits controls the extent (and hence the sensitivity) of the *Iso value* slider.

5. The *Show spatial peak SAR* page displayes SAR (specific absorption rate), as calculated using the SA card (see section 10.2.41). It also displays the cube at the position of maximum SAR when applicable.

6. The *Show UTD rays* page controls the visibility of UTD rays. The rays are not stored by default as the output file can quickly become very big. It must be explicitly activated at the UT card (see section 9.2.41). The *Select* tab is used to select specific rays. Each ray group includes all rays that start and end at the same point. *Ray type* limits the display to certain types of rays. Finally the *Field type* facilitates selecting rays obtained during specific calculations — it is, for example, possible to hide all rays calculated when determining the coupling between the UTD and MoM regions. The *Far field crop* slide bar can be used to restrict far field points to within the geometry display.

   POSTFEKO includes the functionality to display the relative amplitude of the contributions due to the different UTD rays.

All results can be plotted in dB or on a linear scale as selected in the *Scale* group. It is possible to add an offset and a scale factor to the data. This is applied after converting to dB. Adding a 60dB offset changes, for example, dBV to dBmV. The minimum and maximum values used in the colour legend are specified on the *Scale* group. If *Manual limits* is checked the slider bars in this group are disabled and exact values can be entered. If unchecked the sliders control the extents between the minimum and maximum values in the selected data.

Selecting *Instantaneous* on any of these pages shows the results as a function of ωt. For $\omega t = 0°$ this is the real part and for $\omega t = 90°$ it is the imaginary part of the complex quantity. Animation of results is usually found on the *Options* tab on each of these pages and is usually only available for *Instantaneous* results. Also, the results that can be displayed as arrows can generally only be displayed for *Instantaneous* results.

6.4.2 3D result selection

All result pages contain the result selection group shown in figure 6-3. Select the appropriate frequency from the first list. (Note that a new entry is added to this list each time the solution frequency changes. Thus if solutions were done at 1 GHz, 1.1 GHz and again at 1 GHz, there will be three entries in the list.) The *Solution* list contains an entry for each solution at a given frequency. (For example, if a solution is done with each of two sets of sources at a given frequency, there will be two entries in this list.) The final group

typically contains an entry for each card requesting output of the current data type. (For example, if a model has three FE cards in the input file, there will be three entries here.) 3D results can only be plotted as a function of position. Note that for bi-static RCS calculation a single angular sweep contains multiple solutions.

The *Use continuous frequency* checkbox is enabled if the BOF files contains adaptive (continuous) results. (See Chapter 13.) Check this box and click the *Sampling* button to specify the interpolation frequency at which the results must be displayed. The interpolation frequency can be any frequency in the band, not just the specific sample frequencies.

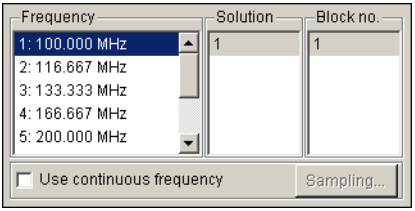


Figure 6-3: The result selector.

6.5 Displaying 2D results

3D displays give a good overview and provide insight, but 2D graphs are required for quantitative display and data comparison. The basic operation of 2D graphs are the same as for 3D displays, but it has extra functionality. First, data from different models can be plotted on the same graph. (On the 3D graph the *Filename* field applies to the entire display. For the 2D graph it applies only to a single series.) Second, POSTFEKO offers to update the data or make a copy of the values when the result (BOF) file changes. (See section 6.5.7.)

The control panel for a 2D graph has a graph settings toolbar on the left and a series control toolbar (used to create, delete and copy series of the specific data type on the graph) at the top. Each series has its own page area where the user selects which data to display. Note that the associated BOF file is specified for each series.

The 2D graphs automatically display results in the appropriate units, for example a plot between 2×10^{11} and 2×10^{11} Hz is displayed between 200 and 400 GHz.

6.5.1 2D result selection

All 2D result pages contain the same selection group (shown in figure 6-3) as the 3D result pages. 2D results can, however, be plotted as a function of frequency, solution number or

block number. Hence multiple elements can be selected from all three lists, but it is only possible to select multiple entries in one block at a time. If any list has multiple entries selected, it determines the independent plot variable. If no list has multiple selected entries, the independent variable defaults to a position variable.

If multiple entries are selected in any of the result lists, the largest common denominator of entries is used in the remaining lists. For example, if a FEKO run used a frequency sweep to calculate only input impedance and calculated the field patterns at one frequency, selecting all frequencies will list the patterns in the third block even though they occur for only one frequency.) Missing entries are just ignored when plotting. Again bi-static RCS calculations behaves differently.

6.5.2 Continuous (adaptive) frequency sampling

As for the 2D case, the *Use continuous frequency* checkbox is enabled if the BOF files contains adaptive (continuous) results. (See Chapter 13.) In this case checking this box immediately sets the independent variable to frequency. However, it is possible to select a single frequency where the results are calculated and plot against another variable such as position. If *Automatically determine sampling depth* on the *Specify sampling* dialog is unchecked, a fixed number of samples can be specified. Thus it is possible to interpolate the result at a number of discrete frequencies.

6.5.3 New 2D graph toolbar

The new 2D graph part of the main toolbar is used to create the different types of graphs. These are only enabled if at least one BOF file is loaded. Each item creates a specific type of graph and it is not possible to plot different quantities (far field, impedance, etc.) on the same graph.

1. *Add a currents and charges graph* plots segment currents and charges. Selecting specific entries under label can significantly reduce the number of entries under segment, making it simpler to find the correct segment.

2. *Add a source data graph* plots source related information, such as input impedance and losses. An arbitrary load can be subtracted with the *Subtract loading* field. If *Impedance* is selected, the specified value is subtracted from the impedance as if it is in series with the input impedance. If *Admittance* is selected, the value is subtracted from the admittance as if the load is parallel to the input impedance. The *Reference* impedance is used for S_{11} and VSWR calculations.

3. *Add S-parameter graph* plots S-parameters calculated using the SP card.
4. *Add a power data graph* plots the active power, loss power and efficiency of the selected source.
5. *Add a non-radiating network graph* plots the impedances, voltages and currents at the ports of transmission lines.
6. *Add a Far Field graph* plots the radiated fields.
7. *Add a Near Field graph* plots graphs of near field values.

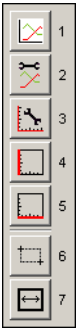
6.5.4 Series manipulation toolbar

This toolbar is used to control the series information for each graph.

1. *Delete the current series* This button will remove the current series from the graph. The currently selected tab determines which series is removed.
2. *New Series* adds another default series to the graph.
3. *Copy Series* copies the current series. This would be identical to creating a new series, and then modifying all the series settings to be the same as the current series.
4. *Copy to value series* creates a new series which is a value series of the current series. (See section 6.5.7.)
5. *Convert to value series* converts the current series to a value series. When a series is copied or converted to a value series the original data is used and any offset or scaling factor must be applied again to the new series.
6. *Perform calculations on series* opens a dialog where calculations can be performed on a series or between multiple series. When using this, the syntax is the same as the syntax for PREFEKO when dealing with variables. For example, to add two series together, the syntax would be *#series_1 + #series_2*. The result of this computation is saved as a value series. (Note the #-character preceding the name of the series as given in the tab on the control panel.) If the *Limit operations to the common region* is checked the calculation is only done in the region where all series have values. (Interpolation between values is done.) If unchecked, a value of 0 is assumed where a series is used outside its range.
7. *Rename series* opens a dialog that can be used to change the name of a series.

6.5.5 Graph settings toolbar

This toolbar, located on the far left when a 2D graph has focus, is used to control the settings of the graph and how the series are displayed.



1. *Series data* opens a panel where the user controls the data for each series. Typical options that would be available on this panel include field components for near fields or which part (real, imaginary, magnitude or phase) of the data to plot. The *Unwrap phase* option should only be used for phase calculations. If it is checked, POSTFEKO tries to avoid 360° jumps by adding integer multiples of 360°.

2. *Series settings* opens a panel that is used to specify graph settings with respect to each series, such as plot colour, markers etc. This panel is also used to specify offsets and scaling factors.

3. *Graph settings* panel controls the global graph settings, such as *Polar* / *Cartesian*, graph title and footer information. For *Polar* graphs checking *Vertical* displays the graph with 0° at the top rather than to the left. If *Mirror* is checked, the angle runs clockwise rather than counter-clockwise.

4. *Left axis* panel controls all the settings for the left axis. These settings include linear/log/dB selections, normalisation and captioning. If *Add unit* is unchecked POSTFEKO does not try to a add a unit — it is then the user's responsibility to do so in the axis caption.

5. *Bottom axis* panel controls all the settings for the bottom axis. These settings are very similar to the left axis.

6. *Zoom box* activates the mouse to specify a zoom window. The panel contents do not change when this button is pressed. Zooming in does not reset the precision of the labels on the axis tick marks. It may be necessary to increase the precision of the labels. To do this use the axis settings panel.

7. The *Reset zoom* button is only visible when zoomed in, and restores the graph to the full dimensions of the axis specifications when pressed.

Note that the graph titles, the axis captions and the series legends each use a check box *Automatic update*. If this is checked, POSTFEKO will attempt to use the correct text in the associated field. (This is determined from the selections on the control panel.) If the text field is changed (this happens only when the field looses focus or the user presses <Enter> while typing in the field), the update box is unchecked automatically. While the box is unchecked, the text is not updated. If the selection is changed while this is the case, the text field will be invalid until one checks the update box again (or edits the text manually).

6.5.6 Zooming / rotating / panning the model

The user is often interested in inspecting details (like close to resonances) of the 2D graph. Thus POSTFEKO allows zooming in on a specific part of the graph. (Note that zooming is not supported for polar plots or Smith charts.) To zoom in to an area, click the *Zoom box* button on the left of the control panel and click and drag with the left mouse button at the required location. (Alternatively, hold <¡>Shift¿ while clicking and dragging.) While the mouse is dragged, a red dotted line indicates the are that will be enlarged once the mouse button is released.

After zooming in on the graph a new *Reset zoom/pan* button is shown below the *Zoom box* button. Click this to return to the default view. Note that when autoscaling for an axis is enabled, the settings are adjusted while zooming (for example, the number of digits in the labels may be increased). When autoscaling is off, this is not done automatically but left to the user.

Clicking and dragging while hoding down <Ctrl> pans (moves) the graph. This is particularly useful in connection with zooming, so that the visible window of an enlarged view of a plot can be moved around.

6.5.7 Value series description

A *value series* is a data series that stores its own values, rather than referring to a result file. The series is simply stored as a series of numbers, and does not necessarily relate to any specific type of data. In contrast, when data is read from a BOF file, POSTFEKO can determine the data type.

A user may wish to convert a series to a value series when the integrity of data in a BOF file is not ensured — for instance, a user may want to store the graph of the input impedance of an antenna versus frequency for a few iterations of the antenna design. The same model is used for each iteration of the design. In such a case, the input impedance is plotted and converted into a value series before the PRE file is modified and the problem is re-simulated.

In the 3D view, all results are updated immediately if the result changes. For the 2D, however, the user is prompted to *Reload* or *Serialise*. If a series is serialised, it is converted to a value series which no longer depends on the result file. (This also implies that the series cannot be modified in a way that requires the original result. It is possible, for example, to switch from dB to linear as this requires only the current values, but it is not possible to switch between electric and magnetic fields.) The control panel that referenced the result file is then replaced with a table of data values. If the update option is selected, POSTFEKO will load the new information from the BOF file and update the graph.

6.5.8 PFG graphs and importing GraphFEKO WFG graphs

Any 2D graph can be saved to a PFG graph. Note that this is a snapshot of the graph at the save time — the PFG graph stores the result data rather than a reference to the applicable BOF file. If the graph is modified afterwards, the save command will save the modified graph as part of the session — the PFG graph will remain unchanged unless it is explicitly saved again. Finally, it should be noted that loading a PFG graph loads it into the active session. The PFG file can only be updated by explicitly exporting the graph again.

If an existing GraphFEKO graph (WFG file) is imported into POSTFEKO (using *File* → *Import* → *Graph*), GraphFEKO is called in the background to convert the graph to a PFG graph. This is only possible on MS Windows. It is also possible to open the WFG graph in GraphFEKO and export it to a PFG graph (using *File* → *Export POSTFEKO graph* in the GraphFEKO main menu). Exporting the graph while working in GraphFEKO could yield a different number of subdivisions than using the import menu in POSTFEKO, but this is the only difference in the two approaches.

When exporting a dB graph, a power quantity is assumed. The dB graph in POSTFEKO will be correct, but if one then change this to a linear scale in POSTFEKO (or look at the raw values in the data panel), all field quantity values will be a factor of 2 to small. This is due to the fact that GraphFEKO does not store this information — since it does not allow linear/dB convertions it does not use this info again once the graph is created.

6.5.9 Loading and saving raw data

POSTFEKO saves all the displays and their settings to the session file. This means that no graphs or images need to be explicitly saved to be reproduced. All that POSTFEKO requires is the FEK, BOF and PFS files to reproduce all result visualisation. In some cases, for information sharing purposes, it may be desirable to save the data itself. Select *File* → *Export* → *Data*. This option exports the current 2D view to a text file that can be read by external programs, such as MATLAB. Note that this exports the raw data (without scale factors or offsets). If it is required to save the scaled data, one must do a series calculation and then export the resulting value series.

POSTFEKO can also read data in this format using *File* → *Import* → *Data*. POSTFEKO queries the user if the data should be imported into a new or an existing graph. POSTFEKO also queries the dB conversion factor for the data (20 for field quantities and 10 for power quantities) and if the data is already in dB. The data must be in columns separated by one or more spaces (no commas). The data in the second and higher columns are plotted against the first column. The table below shows an example of a text file, with data for two lines, ready for import into POSTFEKO. The first line consists of the bottom axis caption (associated with the first column) followed by a legend for each of the remaining data columns. Note that the captions must be in the first row and each caption

must be between quotation marks. If no caption row is present, POSTFEKO will still import the data (first column gives the position on the bottom axis, subsequent columns that on the left axis). If any column (except for the first column) does not have the same number of data entries as the first column, the empty data points must be represented by the text NAN.

```
"Frequency [MHz]"  "mag{S_11}"   "mag{S_21}"
1.00000E+02        -4.74462E-01  -2.83472E+01
2.00000E+02        -1.46055E+00   2.66604E+01
3.00000E+02        -7.46769E+00   2.06073E+01
4.00000E+02        -2.19462E+00   NAN
5.00000E+02        -6.12814E+00   NAN
```

For Smith chart data, the first column must contain the phase (in degrees) and the next columns the amplitude.

6.6 Exporting images 3D / 2D

Use *File* → *Export* → *Image* to export the current view to an image file. Currently BMP, JPG, PNG, XPM and EPS files are support. The EPS files are vector based, but for the 3D views it effectively uses bitmaps of the rendered screen. It is important to ensure that the entire display is visible when exporting it to an image. If another window covers part of the images, it will also cover the exported image.

It is also possible to copy the figure to the MS Windows clipboard, and to export animated (AVI) files from 3D view animations.

6.7 Polarisation

Far field polarisation can be calculated in a number of polarisations. Since some of these are less familiar, they are summarised here (see also figure 6-4):

- *Total:* The total value independent of the polarisation.
- *Vertical:* The vertical (or ϑ) component.
- *Horizontal:* The horizontal (or φ) component.
- *LHC:* The left hand circularly polarised component. The polarisation vector rotates counter clockwise when viewed, at a fixed position, in the direction of propagation.
- *RHC:* The right hand circularly polarised component. The polarisation vector rotates clockwise when viewed, at a fixed position, in the direction of propagation.

- *Z (+45 deg):* When viewed in the direction of propagation, the $\vec{e}_\vartheta$ unit vector points downwards and $\vec{e}_\varphi$ to the left. The unit vector for Z-polarisation is then $\vec{e}_Z = (\vec{e}_\vartheta + \vec{e}_\varphi)/\sqrt{2}$ which lies along an axis rotated +45 degrees from horizontal (in a counter clockwise direction) — coinciding with the direction of the diagonal line of the Z.
- *S (-45 deg):* Here $\vec{e}_S = (\vec{e}_\vartheta - \vec{e}_\varphi)/\sqrt{2}$ which is rotated by -45 degrees from horizontal and lies in the direction approximated by the diagonal of the S.

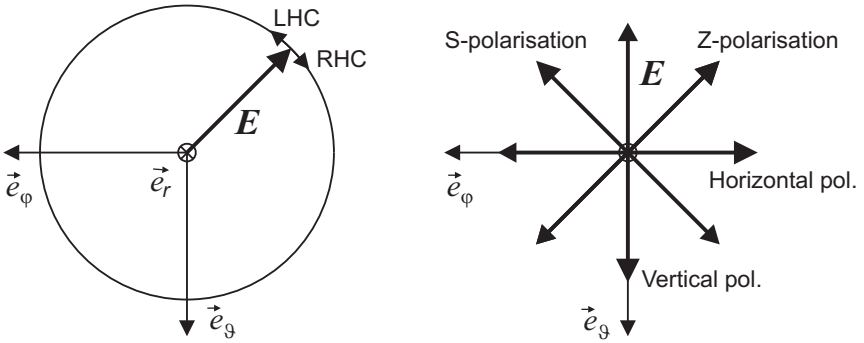


Figure 6-4: Polarisation orientation as viewed, at a fixed position, in the direction of propagation.

6.8 Viewing the output file

The text OUT file contains the FEKO results in human readable format. It is also the only place where one can see the text of the warning and error messages. The OUT file is displayed by clicking the *View the OUT file* button on the main toolbar. It is possible to search for a specific phrase by entering it in the *Phrase* fields and clicking *Search*. Selecting a word in the *Keyword* drop down list enters the phrase associated with that keyword.

If only one model is loaded or a 3D visualisation window (which displays a single model) has focus, the OUT file of the associated model is opened. Otherwise POSTFEKO will prompt for a model. Note that the OUT file listing is not updated automatically when the OUT file changes. The viewer must be closed and reopened to update it.

6.9 Advanced / troubleshooting features

6.9.1 Database browser

The database browser is used to list and control the models in the current session. It is not shown by default and can be activated by selecting *View* → *Database browser* from

the menu. It lists the data that was calculated for each model in the session as well as which data is currently cached (in memory). Files can be removed and / or unloaded by right clicking on them and selecting the appropriate action from the pop-up menu.

6.9.2 Help menu

The *Getting started manual* and *User's manual* can be opened from the *Help* menu. These are PDF documents which requires that a valid PDF viewer is specified on the *Options* → *Settings* dialog.

7 The preprocessor PREFEKO

7.1 Description

The surface of the structure to be analysed with the program FEKO, has to be subdivided into elementary surfaces (in this case triangles). Wires have to be subdivided into segments. The mesh size is dependent on the wavelength in the medium surrounding the structure. The program PREFEKO can do all the meshing. It automatically generates the geometric data, in the form required by FEKO, from the data given by the user. The mesh density is controlled by a couple of parameters. PREFEKO also imports mesh geometry, for example constructed in CADFEKO, integrating it with the final FEKO input file.

This section describe the principle workings of the program. The user first defines the location of points in space with the DP card. Structures are then defined in terms of these points. For example, two points may be joined to form a line (BL card), or four points for a parallelogram (BP card).

7.2 Running PREFEKO

If, for example, a file `example.pre` has been created using a text editor, PREFEKO is started using the following command:

```
prefeko example
```

After successful execution a file `example.fek`, is created. This is the input file for FEKO. The program PREFEKO allows a number of options, which are mainly used for debugging purposes. Entering `prefeko` without arguments will give an overview of the syntax and supported options. If, for example, the argument `--fek-format x` is specified, PREFEKO creates a `*.fek` file using the $\texttt{x}^{\text{th}}$ file format.

The option `--ignore-errors` makes errors non-fatal, i.e. PREFEKO continues with processing after encountering errors. This can result in more errors as a consequence of the first one, but it could also be useful to see all geometry modelling errors at once, and not always the first one only.

PREFEKO also allows the definition of variables (see the next section) from the command line, for example calling PREFEKO with

```
prefeko filename #variable1=value1 #variable2=value2 ...
```

sets the variables `#variable1` and `#variable2` such that it may be used in the `*.pre` file. It is a good idea to use the `!!print_to_out` command to write these variables to the output file in order to keep a record of their values. Note that in UNIX shells the `#` character is a special character and needs to be escaped, for example,

```
prefeko filename \#variable1=value1 \#variable2=value2 ...
```

7.3 Symbolic variables

Instead of using numerical values in all the different cards, it is possible to use predefined variables. The name of a variable always consists of a #-sign followed by a string consisting of the characters a-z, A-Z, 0-9 and the special character _. The following are valid variable names, #height, #a, #STARTINGFREQUENCY, #a_1 or #P5_7f, while the following are not valid: #a?1 or #value2.1. There is no distinction between upper and lower case characters. For example, #a and #A are interpreted as the same variable.

Expressions and functions may be used when defining variables, so that direct calculations can be carried out. The variables have to be defined before they can be used in the respective cards. It is possible to use expressions like 2*#radius in the input fields subject to the maximum allowed length (10 characters for real values, 5 characters for integer values). For larger expressions additional variables have to be defined.

A variable is defined in the following way:

```
#2pi = 2*#pi
#vara = 1 + sqrt(2)
#varb = #vara * 2.3e-2 * (sin(#pi/6) + sin(rad(40)) + #vara^2)
#sum = #vara+#varb
```

Note that the # sign has to appear in the first column. Variables may also be defined from the command line as described in the previous section.

In addition to plain variables also arrays with integer indices are supported like #a[5] or, more complex, #am_0[3*#i+ceil(#r^2)]. The expression between the square brackets must evaluate to an integer number, which can also be negative. The implementation of using arrays is such that they do not need to be allocated, however they need to be initialised. So for instance after having used the instructions

```
!!for #i = 10 to 20
#array_a[#i] = 3*#i-10
#array_b[-#i] = 0
!!next
```

one will be able to use #array_a[10] or #array_a[17] or also #array_b[-12] in other expressions. But trying to use for instance #array_a[5] or #array_b[0] will result in an error message that an undefined variable is used.

On the right hand side of any expression, variables that have already been defined can be used in conjunction with any of the following functions:

()	brackets
+	addition
−	subtraction

*	multiplication
/	division
^	powers, for example 2^3=8
SIN	sine (argument in radians)
COS	cosine (argument in radians)
TAN	tangent (argument in radians)
COT	cotangent (argument in radians)
ARCSIN	arcsine (in radians)
ARCCOS	arccosine (in radians)
ARCTAN	arctangent (in radians)
ATAN2	this function has two arguments `atan2(#y,#x)` it yields `arctan(#y/#x)` in the range $-\pi \ldots \pi$
ARCCOT	arccotangent
SINH	hyperbolic sine
COSH	hyperbolic cosine
TANH	hyperbolic tangent
SQRT	square root
LOG	logarithm to the base 10
LN	natural logarithm
EXP	exponential function
BESJ(n,x)	Bessel function $J_n(x)$ of integer order $n \geq 0$ and real argument x
BESY(n,x)	Neumann function $Y_n(x)$ of integer order $n \geq 0$ and real argument x
BESI(n,x)	Modified Bessel function of the first kind $I_n(x)$ of integer order $n \geq 0$ and real argument x
BESK(n,x)	Modified Bessel function of the second kind $K_n(x)$ of integer order $n \geq 0$ and real argument x
ABS	absolute value
DEG	convert radians into degrees
RAD	convert degrees into radians
STEP	step function, i.e. $\text{STEP}(x) = 0$ for $x \leq 0$ $\text{STEP}(x) = 1$ for $x > 0$
CEIL	smallest integer value that is equal to or greater than the argument
FLOOR	largest integer value that is equal to or smaller than the argument
MAX	returns the largest of the two arguments — called as `max(#a,#b)`
MIN	returns the smallest of the two arguments — called as `min(#a,#b)`
FMOD	this function also has two arguments `fmod(#a,#b)` and returns the remainder of the division `#a/#b`

RANDOM This function returns a random value in the range 0...1. If the argument X of RANDOM() is -1, then a random number is returned. For any other argument X in the range 0...1 this value is used to set the seed, and then a random number is created using this seed. (Using the same seed allows one to create a deterministic and reproducible random number series). If "RANDOM(-1)" is called before any seed is set in the `*.pre` file, then the returned values are random and not reproducible. (The internal seed is used based on the time when PREFEKO is executed).

X_COORD This function returns the x coordinate of a point previously defined by a DP card. The name of the point, in quotation marks, is passed as an argument to the function, for example

```
DP   PNT01                        1.234      0.4567     #z
#x = x_coord("PNT01")
```

will set the parameter `#x` equal to 1.234

Y_COORD Returns the y coordinate of a previously defined point similar to the function X_COORD discussed above.

Z_COORD Returns the z coordinate of a previously defined point similar to the function X_COORD discussed above.

The FILEREAD function reads a numerical value from an arbitrary ASCII file. The general syntax is

```
fileread("Filename", Line, Column)
```

and contains the filename, the line number to read from and the column to read. (The data in the respective columns of any line are separated by one or more spaces or tab characters.) For example with a file containing

```
Frequency in MHz      Re(load) in Ohm    Im(load in Ohm)
   100                    22.54               -12.56
   150                    25.07                -6.54
   200                    27.42                 0.23
```

the frequency and loading can be imported directly from the file

```
#numfreq = 3    ** Number of frequencies
!!for #i = 1 to #numfreq

** Define the frequency (conversion from MHz to Hz)
#freq = 1.0e6*fileread("datafile.dat", #i+1, 1)
FR   1    0                   #freq
```

```
** Define the load
#Zr = fileread("datafile.dat", #i+1, 2)
#Zi = fileread("datafile.dat", #i+1, 3)
LZ   0                          #Zr       #Zi

** Computations ...

!!next    ** End of frequency loop
```

In addition to these functions PREFEKO allows the use of logical operations. It supports the function NOT() — which returns TRUE if the argument is FALSE and FALSE when the argument is TRUE — and the delimiters `>`, `<`, `>=`, `<=`, `=`, `<>`, AND and OR. When boolean operations are applied to variables, a value of 0 is taken as FALSE and everything else is interpreted as TRUE. Similarly in the result of a logical operation FALSE is mapped to 0 and TRUE to 1.

PREFEKO also supports a logical function DEFINED(`#variable`) which returns TRUE if a the variable `#variable` has been defined, and FALSE if not. This is useful in `*.pre` files where during OPTFEKO, TIMEFEKO or ADAPTFEKO runs certain variables are inserted into the header of the file. One can now process and view such `*.pre` files. For example in a `*.pre` file which will be optimised with respect to the variable `#a`, one may use

```
!!if (not(defined(#a))) then
#a = 200.0e-3
!!endif
```

The order of precedence is (lower levels are evaluated first) is

OR
AND
`=` and `<>`
`>`, `<`, `>=` and `<=`
`+` and `-`
`*` and `/`
`^`
`+` and `-` (when used as sign)
function calls
single number, expressions in brackets

Some variables are predefined in PREFEKO, but may be overwritten by re-assignments. These are:

Name	Value	Description
`#pi`	$3.14159265358979\ldots$	The constant π
`#eps0`	$8.85418781761 \cdot 10^{-12}$	Dielectric constant ε_0 of free space
`#mu0`	$4\,\pi \cdot 10^{-7}$	Permeability μ_0 of free space
`#c0`	$\frac{1}{\sqrt{\mu_0\,\varepsilon_0}}$	The speed of light in free space
`#zf0`	$\sqrt{\frac{\mu_0}{\varepsilon_0}}$	The intrinsic impedance of free space
`#true`	1	Used for logical true
`#false`	0	Used for logical false

There are three other special variables `#!x`, `#!y` and `#!z` which are very useful for the connection of complex wire structures. The three variables specify the Cartesian coordinates of the end point of the wire segment most recently defined. This enables the correct and easy connection of a straight wire to a curved length of wire, as the next extract from an input file demonstrates:

```
CL        .....
DP        A                    #!x      #!y       #!z
#z = #!z + 0.5
DP        B                    #!x      #!y       #z
BL        A    B
```

The following example demonstrates the use of variables.

```
** A dielectric sphere in the field of an incident wave

** Define the variables
#r = 1          **   Radius of the sphere
#betrad = 1     **   Electrical size of the sphere
#epsr = 15      **   The relative dielectric constant
#maxlen = 0.7   **   The maximum edge length

** Define segmentation parameters
IP                                      #maxlen

** The corner points
DP     A                       0        0         0
DP     B                       0        0         #r
DP     C                       #r       0         0

** Select the medium
ME     1    0
```

```
** Generate an eighth of the sphere
KU     A    B    C             0         0         90        90        #maxlen

** Use symmetry in all three coordinate planes
**   yz-plane: ideal electrically conducting plane
**   xz-plane: ideal magnetically conducting plane
**   xy-plane: only geometrically symmetric
SY   1    2    3    1

** End of the geometry
EG   1    0    0    0    0

** Assigning the dielectric's properties
DI                           #epsr     1.0

** Incident plane wave excitation
#freq = #betrad * #c0 / (2*#pi*#r)
FR   1    0                  #freq
A0   0         1    1         1.0       0.0      -180.0

** Near fields along the z axis
FE   1    1    1    25   0    0.0       0.0      -1.98     0.0      0.0      0.04
FE   4    1    1    50   0    0.0       0.0      -0.98     0.0      0.0      0.04
FE   1    1    1    25   0    0.0       0.0       1.02     0.0      0.0      0.04

** End
EN
```

The use of variables makes the investigation of structures with varying geometry (e.g. variable distance of the antenna in front of a reflector) an easy process, because only one variable needs to be changed. It also allows FOR loops and IF conditions.

7.4 FOR/NEXT loops

Some cards in FEKO implicitly use loops (such as when an FR card with multiple frequencies is used). This, however, does not always offer the flexibility which one may require, for example, to change the material parameters inside the loop. Another example would be the use of a loop to create a complex geometry.

For completely general loops, PREFEKO allows the construct

```
!!for #var = #start to #end step #delta

!!next
```

where a simple example would be

```
** Loop for the relative permittivity
!!for #eps_r = 1 to 5 step 0.5

** Set material parameters
GF   0                          #eps_r    1

** Compute fields etc.
FE

** End of loop
!!next
```

The syntax requirements of FOR/NEXT loops are:

- The `!!` characters must be located in the first two columns of the line. This is followed by a number of optional spaces and the keyword `for` (it is not case sensitive, so also `FOR` or `For` are accepted).
- The keyword `for` is followed by the name of the loop variable (starting with `#`). The variable name is terminated by a space or the `=` sign.
- Next follows an expression for the initial value of the loop (a constant, variable or formula, see the example below).
- This is followed by the keyword `to` and the terminating value of the loop variable (again a constant, variable or formula).
- The default increment of the loop variable is 1, but it can be changed by using the keyword `step` followed from an expression. Negative increments are allowed.
- The loop is terminated by a line of the form `!!next` (spaces are allowed between `!!` and `next` but not before the `!!`). All instructions and input cards between `!!for` and `!!next` are evaluated repeatedly inside the loop.
- Several loops can be nested as shown in the example below.

A more complicated example, illustrating some of the points above, is as follows

```
#end = 3+sin(4)
!!for #x1 = sqrt(5) + 2*3 to 2*#end step -#end/10
!!  for #x2 = 1.23 to 2*#x1   ** this is the inner loop

#x3 = #x1 + #x2
DP ....
.... (more commands)

!!  next
!!next
```

7.5 IF/ELSE/ENDIF constructs

This construct is used to allow different control cards under different conditions. The syntax requirements of IF/ELSE/ENDIF constructs are:

- The `!!` characters must be located in the first two columns of the line. This is followed by an arbitrary number of spaces, the expression to evaluate evaluated and the keyword `then` (it is not case sensitive, `THEN` or `Then` are also accepted).
- The block is terminated by a line of the form `!!endif` (again spaces are allowed between `!!` and `endif` but not before the `!!`).
- An optional line of the form `!!else` (again, the `!!` must be in the first two columns and spaces are allowed before the keyword which is not case sensitive).
- All instructions and input cards between `!!if` and `!!endif` (or `!!else` if it is present) are processed if the expression is TRUE. If it is present, all lines between `!!else` and `!!endif` are processed if the expression is FALSE.

Examples, illustrating some of the points above, are as follows

```
!!if #a > 5 then

...

!!endif
```

or

```
#l = (#a+5 > 21) and (#a < 100)
!!if ( (3*#a+5 >= #x/2) and not(#l) ) then

...

!!else
!!  if (sin(#x/10) > 0.5) then

...

!!  else

...

!!  endif
!!endif
```

7.6 Symbolic node names

When defining or using node names, simple variable names of the form `A#i` are allowed. The algorithm is that if a hash sign is found in a node point name, this hash sign and everything that follows is interpreted as a variable string, evaluated and rounded to the nearest integer. Thus if we have `#k=15` and use or define a point `P#k` then this is equivalent to using `P15` as point name. The length of the node name string (before and after expansion) is still limited to 5 characters.

For instance, it would now be possible to define the points `P1` to `P20` inside a loop

```
!!for #k = 1 to 20
DP   P#k            .....
!!next
```

and use these either individually or inside another loop.

7.7 PRINT and EXIT commands

PREFEKO also supports the command `!!exit` to stop execution and the commands `!!print` and `!!print_to_out` to print strings (enclosed in double quotes) and floating point numbers to the screen and `*.out` file respectively. The `!!print` and `!!print_to_out` commands accepts multiple arguments separated by commas. For example,

```
!!print "2*#b = ", 2*#b
!!if #a < 2*#b then
!!  print "The value of #a is too small:", #a, " (exiting now)"
!!  exit
!!endif
```

will print a warning and exit if the variable `#a` is smaller than two times variable `#b`. The line

```
!!print_to_out "This run was done with #b = ", #b
```

will print the value of `#b` to the `*.out` file at the position where it appears in the `*.pre` file.

8 The FEKO solution kernel

8.1 Introduction

The program FEKO does the actual field calculation. Input and output are done using files. The program indicates, on screen, how far the calculation has progressed.

8.2 Running the FEKO kernel

8.2.1 Running the sequential version

Normally users are advised to run the FEKO kernel directly from the GUI components CADFEKO, EDITFEKO or POSTFEKO. Once a session or model has been loaded, the sequential FEKO solver can be started from the *Run* menu, there select *FEKO* (as a shortcut key `Alt+4` can be used). Note that running the FEKO kernel will also automatically execute the pre-processor PREFEKO if the `*.fek` file does not yet exist, or if the corresponding `*.pre` or `*.cfm` files are newer than the `*.fek` file.

When the FEKO kernel is not executed from within the GUI, it can be started in a command window (on a Windows PC) or a shell (in UNIX) by executing the command

```
runfeko example_08
```

where `example_08.pre` must be an existing input file. RUNFEKO executes PREFEKO if the `*.fek` file is missing or older than the `*.pre` file and then executes the appropriate FEKO solver (sequential or parallel etc.) RUNFEKO accepts the following optional parameters (see also section 8.2.2 for additional options to launch and control the parallel version of the solver, or section 8.2.3 for remote launching of FEKO)

`--execute-prefeko`	Always execute PREFEKO — even if `*.fek` file exists is newer than the `*.pre` file.
`--use-mpich`	Force parallel UNIX versions to use the MPICH version of FEKO, even though the default could be ScaMPI or Score or another MPI implementation.
`--version`	Print the version information and then exit.
`--priority x`	The value `x` specifies the CPU usage priority of the FEKO run: 0 = idle, 1 = below normal, 2 = normal, 3 = above normal and 4 = high. If not specified, the default is 2. This option might not be available for specific systems or specific FEKO versions, then it is just ignored.
`--prefeko-options ...`	All options following this (if one is used, up to the next `--xxx-options`) are passed to PREFEKO.

`--feko-options ...`	All options following this (if one is used, up to the next `--xxx-options`) are passed to FEKO.
`--adaptfeko-options ...`	All options following this (if one is used, up to the next `--xxx-options`) are passed to ADAPTFEKO.
`-d`	Debug mode with some extra output (can be used to troubleshoot error)

The optional command line parameters for FEKO (specified after `--feko-options`) are

`--check-only`	If this option is used, FEKO processes and checks the geometry, but does not start a solution. This is useful to test an input file on a local PC before submitting it to a parallel computer.
`-e ENV=value`	This has the same effect as starting FEKO with the environment variable `ENV` set to `value`. More than one `-e ...` argument is allowed.

8.2.2 Running the parallel version

In order to use the parallel version of FEKO from the GUI, one first has to configure the hostnames and number of processes which shall be used for each node. (This will be setup initially during the installation, so one has to re-configure only if there are changes.) To do this configuration, open the *Run* menu, then *Solution options*, there on the top click on the *FEKO* tab and under *Parallel execution* click the *Configure* button. Then a dialog as shown in figure 8-1 will open. There the hostnames and number of processes to be started on each host must be entered. Usually one would run one process per CPU, which determines the number of processes for each host, for example 2 processes for a dual-board machine. One may also use this for a crude load balancing, running more processes on hosts with faster CPUs or more memory. The *Add machine* button allows to add more nodes.

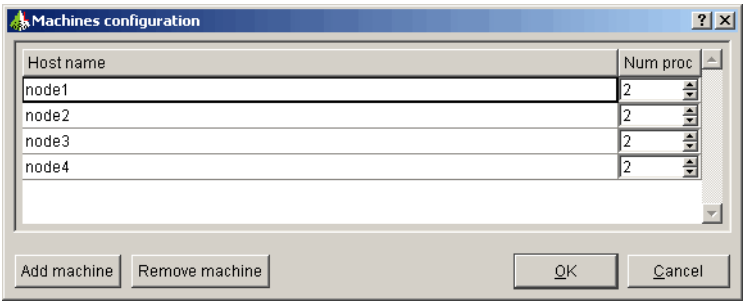


Figure 8-1: Dialog for the parallel host configuration

A special note for parallel Windows PC clusters: FEKO must be installed in the same location on each host, and it is recommended that the parallel job is started from a PC that forms part of the cluster and that this host is listed first.[2]

When the user clicks *OK* on this window, the hosts are saved to a file `machines.feko` in the directory specified by the environment variable `FEKO_USER_HOME`. This file is then used in the actual parallel process launching.

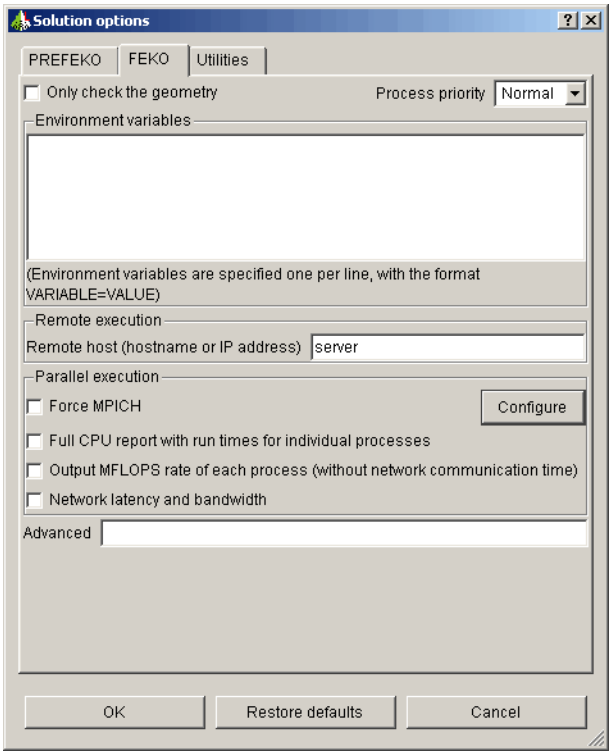


Figure 8-2: Dialog for setting FEKO solution options

On the main *Solution options – FEKO* panel as shown in figure 8-2 some further settings can be made. If the *Output MFLOPS rate ...* and *Network latency and bandwidth* options are selected, FEKO will also print a table giving the performance of the various nodes. It is recommended that this is used during setup to ensure an optimum configuration.

[2]It is possible to launch the job without including the local machine. The `*.fek` input file must then be located on the first PC in the list and the `*.out` and `*.bof` output files are created on this PC — both in the exactly same directory as the project directory on the local machine. It is *the user's responsibility to transfer the files between the local machine and the first machine in the list if these are not the same*, or one can also use remote parallel launching where FEKO does this copying explicitely, see section 8.2.3 for details.

(These checks are repeated each time FEKO calculates the solution, so it may require a significant amount of time if the test file contains multiple frequencies. One would not keep these options selected after the initial setup, except for debugging purposes.)

The user can also set the target priority of the FEKO run in this window. Setting the priority below normal will allow the user to continue with other interactive work. However, all machines in the cluster operate at the speed of the slowest node, so starting other CPU intensive jobs on one of the nodes in a cluster is generally not recommended.

After having configured the parallel FEKO version and after having set any possible special options, the parallel FEKO version can be run. To do this, click in the *Run* menu on *Parallel FEKO execution* so that this gets checked. Then any subsequent runs of the FEKO solver until this gets un-checked again will use the parallel version of FEKO.

From the command line (e.g. on a UNIX workstation), the parallel FEKO version is started with

```
runfeko example_08 -np x
```

where the parameter `x` following `-np` gives the required number of processes. In addition to the arguments listed in section 8.2.1, the parallel version accepts the following optional parameters

`-np x`	Start the parallel FEKO version with x processes.
`--machines-file machname`	The file `machname` is the machines file with the node names and the number of CPUs (see below).
`--mpi-options ...`	All options following this (if another `--xxx-options` parameter is used, all arguments before the second `--xxx-options` parameter) are passed to the MPI launcher (e.g. mpirun or mpiexec).

The number of processes to start on each available host is specified in a so-called *machines file* with the general syntax

```
Hostname:Number of processes
```

using a new line for each host. For example, if the user has two hosts with names `host1` and `host2` (this is the output of the UNIX command `hostname`), and 4 and 8 processors respectively, the machines file will be

```
host1:4
host2:8
```

With this machines file, the example with 6 processes given above will run with 4 processes on `host1` and 2 on `host2`.

If only one process will be started on any host, then instead of the entry `host3:1` in the machines file, also the shorter form `host3` can be used.

Such a machine file (the file `mpi/share/machines.feko` under the FEKO installation path `FEKO_HOME`) is automatically created during the installation of the parallel version of FEKO. By default FEKO uses this file. If a different distribution of the processes is required, one can edit this file manually. This is, however, strongly discouraged. The user should rather create a separate machines file with the syntax described above. If this file is, for example `machname`, the environment variable `FEKO_MACHFILE` can be used to force RUNFEKO to use this file instead of the default. The required commands (for the sh shell) are

```
FEKO_MACHFILE=./machname
export FEKO_MACHFILE
runfeko example_08 -np 6
```

Alternatively one may pass the name of the machines file as an argument to RUNFEKO on the command line like

```
runfeko example_08 -np 6 --machines-file ../../mymachines
```

Using RUNFEKO is independent of the respective platforms and MPI implementations (see also the discussion of the environment variable `FEKO_WHICH_MPI` in section 2.8). For very special applications or experienced users it may be necessary to pass additional options to MPI. (In such a case, the appropriate MPI manuals should be considered.) These are added after the argument `--mpi-options`. For example on a ScaMPI cluster (assuming `FEKO_WHICH_MPI=6`), the call

```
runfeko example_08 -np 6 --mpi-options -immediate_handling \
   threaded -smtrace 5-6
```

(all on one line) is interpreted internally and FEKO is executed with the command

```
/opt/scali/bin/mpimon -export env -immediate_handling threaded \
   -smtrace 5-6 /opt/feko/bin/feko.csv example_08 -- host1 4 \
   host2 2
```

(Note that `host1` and `host2` are examples only — the actual information is taken from the machines file.)

8.2.3 Running on a remote host

The FEKO kernel can also be started on a remote host with automatic file transfer. For instance the user can run the FEKO user interface on a Windows PC, but start a sequential or parallel FEKO job directly from this user interface on some other remote workstation or cluster. This remote launching is also cross platform, e.g. one can launch a remote job from a Windows PC on a UNIX workstation.

Please see the detailed installation and setup instructions in the *Getting started manual*. For instance, SSH must be available with public key authentication.

After the setup, using remote launching is simple. On Windows and Linux, this remote launching facility can be used directly from within the GUI components CADFEKO, EDITFEKO or POSTFEKO. There as described above for the parallel launching open the *Solution options* dialog from within the *Run* menu. This is shown in figure 8-2. Enter the hostname or IP adress of the remote host in the input field *Remote host* (in the example in figure 8-2 this hostname is `server`). In order to then actually also use the remote launching, click in the *Run* menu on *Remote FEKO execution* so that this gets checked. Then any subsequent runs of the FEKO solver (either sequential or parallel if *Parallel FEKO execution* is also checked) until this *Remote FEKO execution* gets unchecked again will be launched remotely on the remote host.

In order to use this remote launching facility from the command line, the command

```
runfeko example_08 --remote-host h
```

can be used where the parameter `h` following `--remote-host` gives the hostname or the IP address of the remote host. This command line option of RUNFEKO can be combined with other options, for instance using

```
runfeko example_08 --remote-host h -np 4 --machines-file m
```

would launch a parallel job with 4 processes using the nodes as listed in the machines file `m`, and the parallel job is then launched from the remote host `h` (which is typically some control node of a cluster).

As mentioned initially, the remote launching facility has an automatic file transfer included, so it is not necessary to work on some kind of shared network drive. On the remote host, FEKO will in the user's home directory create a temporary sub-directory with the name `remote_FEKO_job_xxx` (`xxx` is a unique number) and all the FEKO files will be put there for the duration of the FEKO run. After the completion of the remote run, all files will be copied back to the client and this temporary subdirectory on the remote machine will be removed again.

A final note shall be given which applies to remote launching of parallel FEKO jobs:

- If a machines file is specified while launching the job locally, this will also be used on the remote host (i.e. it will be copied over). So one can from the local client configure to run for instance a parallel job on the hosts `node1` and `node2` but launch remotely on some other computer `control_node`. This makes sense when one wants for instance to start a parallel FEKO job from a Windows PC on a Linux cluster.
- When during the remote launching no local machines file is specified (from the GUI it will always be there, but from the command line this can be omitted), then the parallel hosts are found using some default machanism (environment variable `FEKO_MACHFILE`, default location for the file `machines.feko` etc., see section 8.2.2).

It is important to realise that these default options are used as set on the remote host where the parallel job is launched (this is e.g. the control node of a parallel cluster). Thus for instance we will read the `machines.feko` file on the remote host, and not on the local host where jobs are launched.

9 Description of the geometry cards

9.1 Overview of the geometry cards

The following table lists all input cards that are used to create the geometry (i.e. the cards that appear before the EG card in the `*.pre` file[3]). Most of these cards are processed by PREFEKO. For example, PREFEKO processes the BP card and writes the triangle elements to the `*.fek` file as input to FEKO.

Card	Description
`**`	characters used to indicate a comment
`BL`	creates a line
`BP`	creates a parallelogram
`BQ`	creates a quadrangle
`BT`	creates a triangle
`CB`	changes already assigned labels
`CL`	creates a circular line using segments
`CN`	changes the direction of the normal vector
`DK`	creates a dielectric or magnetic eighth of a sphere
`DP`	defines a node point
`DZ`	creates a cylindrical dielectric shell
`EG`	defines the end of the geometry
`EL`	creates a segment of an ellipsoid
`FM`	set parameters related to the MLFMM
`FO`	defines a Fock region
`FP`	set parameters related to the FEM
`HE`	creates a coil from wire segments
`IN`	reads an external *include* file containing mesh information
`IP`	sets the parameter that defines the degree of meshing
`KA`	defines the border of the PO area
`KK`	creates a elliptical conical segment
`KL`	sets the wedges in the PO area
`KR`	creates a planar elliptical element
`KU`	creates a spherical element
`LA`	specifies the label for segments, triangles, polygons, etc.
`ME`	defines the medium
`NU`	creates a NURBS surface from specified control points
`PB`	creates a paraboloid
`PH`	creates a flat plate with an elliptic hole
`PM`	creates a polygonal shape that is meshed into triangles

[3] In general all the geometry cards must appear before the EG card. Exceptions are the IN card (when including `*.pre` files with control commands) and the DP and TP cards (when defining points for the AP card).

PO	applies the Physical Optics approximation
PY	creates a polygonal surface for use with UTD
QT	creates a dielectric or magnetic cuboid (meshed into tetrahedral elements)
QU	creates a dielectric or magnetic cuboid (meshed into cuboidal elements)
RM	remeshing and adaptive mesh refinement
SF	enters a scaling factor, with which all dimensions are multiplied
SY	utilises symmetry in the construction of the geometry
TG	transformation (i.e. translation and rotation) of the geometric structures
TO	creates a toroid
TP	transforms a point
UT	parameters for the uniform theory of diffraction (UTD)
UZ	creates a cylinder for use in the UTD region
WG	creates a parallelogram consisting of a wire grid
ZY	creates a cylindrical element

9.2 Alphabetical description of the geometry cards

9.2.1 ** card

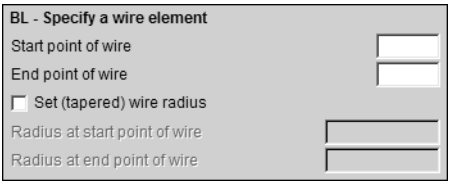
The ** card is not a command, but defines a comment line. Everything that is found in this line is ignored by PREFEKO.

It is possible to add a comment to the end of an existing line or card. For example,

```
** Definition of parameters
#lambda = 1.0          ** Wavelength
#radius = #lambda/2    ** Cylinder radius
#height = 2*#lambda    ** Cylinder height
```

9.2.2 BL card

This card is used to connect two points to form a line, which is then subdivided into wire segments.


BL - Specify a wire element
Start point of wire
End point of wire
Set (tapered) wire radius
Radius at start point of wire
Radius at end point of wire

Parameters:

Start point of wire:	The start point of the wire, previously defined with the DP card.
End point of wire:	The end point of the wire, previously defined with the DP card.
Set wire radius:	If checked, the radius set at the previous IP card is overridden for the current wire. This setting does not affect segments created after this card. Both radius values are in m and are affected by the SF card scaling factor. If only the start radius is specified, the wire will have a constant radius.
Radius at start . . . :	The radius of the wire at the start point.
Radius at end . . . :	The radius of the wire at the end point.

The points have to be defined by a DP card, prior to using this card. The wire radius is set by an IP card preceding the BL card, but can be set locally. Note that by using different radius values for the start and end points of the wire, a tapered wire can be created.

Figure 9-1: Sketch illustrating the use of the BL card

Examples of BL card usage: The BL card can be used to create segmented wires as shown in figures 9-2 and 9-3. In the first example the radius is specified with an IP card and in the second an exaggerated taper is specified at the BL card.

Figure 9-2: Example of a BL card from `demo_BL1.pre`

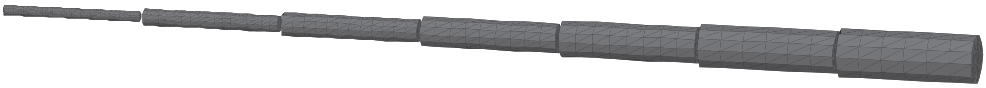
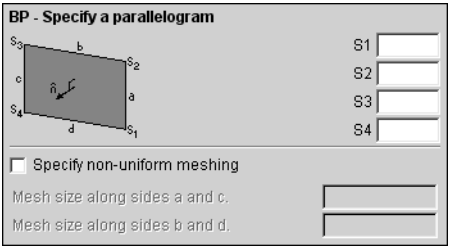


Figure 9-3: Example of a BL card with a tapered radius from `demo_BL2.pre`

9.2.3 BP card

A mesh of surface triangles in the shape of a flat parallelogram can be created with this card. In general, this card is replaced by the PM card. This card should only be used when the user wants to force the very regular meshing that this card produces.



Parameters:

S1, S2, S3, S4: The points S1 to S4 are the four corner points of the parallelogram. These points should have been defined previously with DP cards.

Specify non-uniform . . . : Normally a parallelogram is segmented according to the edge length specified with the IP card. In some cases, e.g. when creating small microstrip lines, it may be desirable to use a finer segmentation in one direction. Check this item if a finer segmentation is required in one direction. The mesh sizes are in m and are scaled by the SF card.

Mesh size along sides a and c: Edges S1–S2 and S3–S4.

Mesh size along sides b and d: Edges S2–S3 and S4–S1.

The points are connected in the order that they appear in the BP card. Thus the user has to ensure that the points describe a parallelogram. If this is not the case, then PREFEKO will abort with the appropriate error message.

The direction of the normal vector ($\hat{n}$) of the subdivided triangles is determined by the right hand rule, through all the corners. This direction is only important when used with the Physical Optics (PO card, section 9.2.31) or with dielectrics (ME card, section 9.2.26) or with the CFIE (CF card, section 10.2.20).

Example of BP card usage:

Figures figure 9-4 and figure 9-5 shows a plate with uniform meshing and a strip with non-uniform meshing respectively, both created with the BP card.

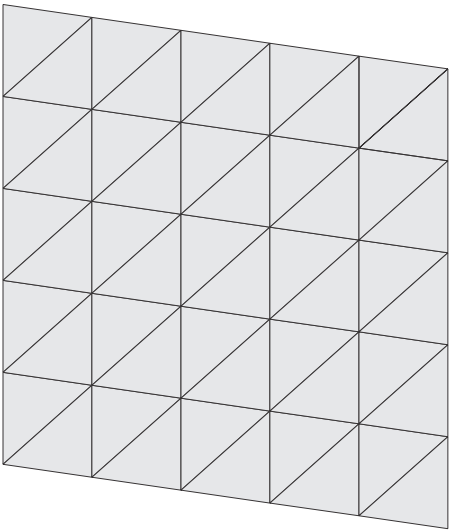


Figure 9-4: Example for the BP card from `demo_BP1.pre`

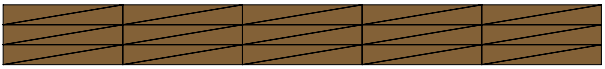
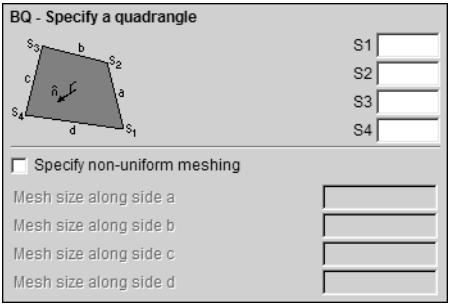


Figure 9-5: Example of a BP card with non-uniform meshing from `demo_BP2.pre`

9.2.4 BQ card

A mesh of surface triangles in the shape of a flat quadrangle can be created with this card. Models constructed using the BQ card can generally be simplified by using the PM card.



Parameters:

S1, S2, S3, S4: The points S1 to S4 are the four corner points of the quadrangle. These points should have been defined previously with the DP card.

Specify non-uniform . . . : Normally a quadrangle is segmented according to the edge length specified with the IP card. In some cases, e.g. when creating small microstrip lines, it may be desirable to use a finer segmentation in one direction. Check this item if finer segmentation is required along any edge. The mesh sizes are in m and are scaled by the SF card.

Mesh size along side a: Edge S1–S2.

Mesh size along side b: Edge S2–S3.

Mesh size along side c: Edge S3–S4.

Mesh size along side d: Edge S4–S1.

The points have to be predefined using DP cards prior to the BQ card and are connected in the order that they appear in the BQ card.

In principal the BQ card can create all types of quadrangles, including parallelograms. The difference is that the BP card creates a regular subdivision.

The direction of the normal vector ($\hat{n}$) of the subdivided triangles is determined by the right hand rule through all the corners. This direction is only important when used with the Physical Optics (PO card) or with dielectrics (ME card) or for the CFIE (CF card).

Examples of BQ card usage:

The BQ card can be used to create the mesh shown in figure 9-6, or also for inhomogeneous meshing as shown in figure 9-7.

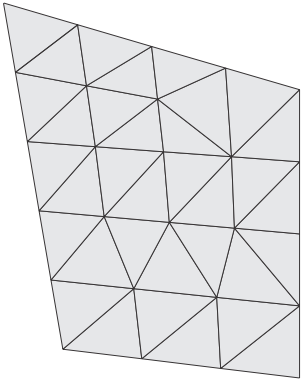


Figure 9-6: Example of a BQ card from `demo_BQ1.pre`

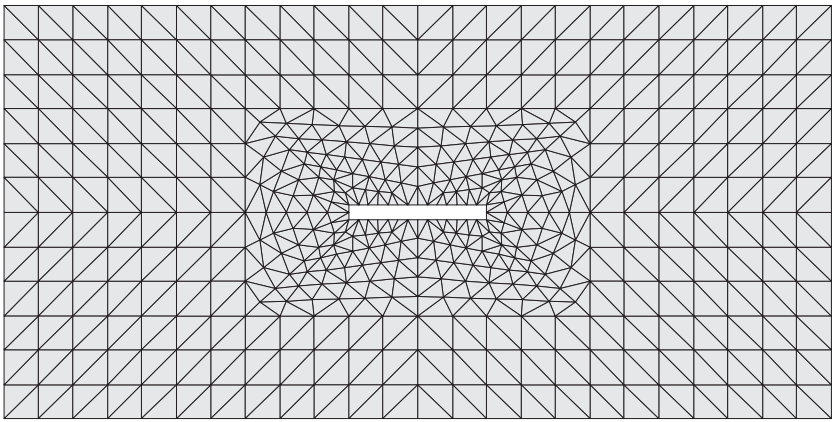
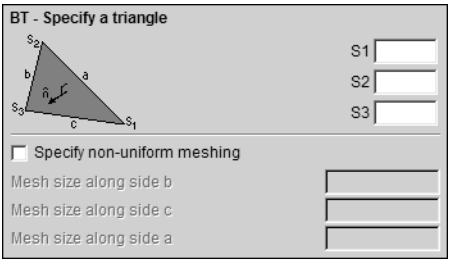


Figure 9-7: Example of a BQ card with an inhomogeneous segmentation from `demo_BQ3.pre`

9.2.5 BT card

A mesh of surface triangles in the shape of a flat triangle can be created with this card. In general, this card is replaced by the PM card.



Parameters:

S1, S2, S3: The points S1 to S3 are the three corner points of the triangle. These points should have been defined previously with the DP card.

Specify non-uniform . . . : Normally a triangle is segmented according to the edge length specified with the IP card. In some cases it may be desirable to use a finer segmentation in one direction. Check this item if finer segmentation is required along any edge. The mesh sizes are in m and are scaled by the SF card.

Mesh size along side b: Edge S2–S3.

Mesh size along side c: Edge S3–S1.

Mesh size along side a: Edge S1–S2.

The direction of the normal vector ($\hat{n}$ of the subdivided triangles is determined by the right hand rule through all the corners. This direction is only important when used with the Physical Optics (PO card) or with dielectrics (ME card) or for the CFIE (CF card).

Examples of BT card usage:

The meshes shown in figures 9-8 and 9-9 were created with BT cards using uniform meshing and non-uniform meshing respectively.

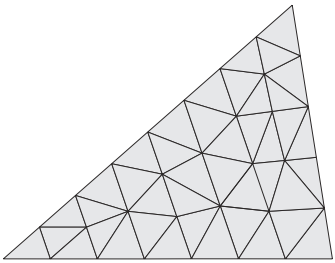


Figure 9-8: Example of a BT card from `demo_BT1.pre`

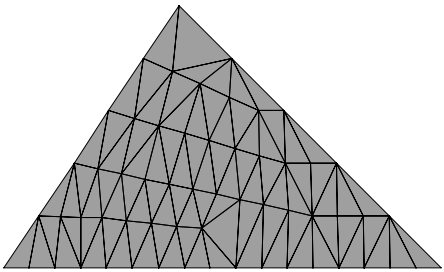
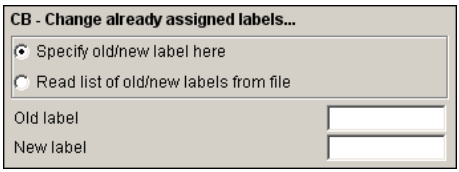


Figure 9-9: Example of a BT card with non-uniform meshing from `demo_BT2.pre`

9.2.6 CB card

This card is used to change or reassign the labels assigned to points, segments, triangles, cuboids, polygons, tetrahedral elements, etc.



Parameters:

Specify old/new label here: This selection allows to specify an old label and a new label in the corresponding input fields.

Read list of old/new labels from file: This selection allows to read a list of old/new label pairs from an external data file. See further details below.

Old label: All the structures with this label are relabelled.

New label: The new label for all the structures with the old label.

Filename: The name of the file used when reading the label list from an external data file.

Renaming labels is especially useful when more labels are created by using symmetry (SY card) or transformations (TG card) or an imported geometry from CADFEKO, and for example, edges or wedges in the PO area are considered or any other properties shall be set by label (e.g. Skin effect). Structures created after the CB card are not affected.

In order to make the renaming of a whole set of different labels simpler, the *Old label* field in the CB card is also supporting wildcards '*' (an arbitrary sequence of characters) and '?' (a single arbitrary character). So for instance to rename all these labels

```
Cube.Face1
Cube.Face2
Cube.Face3
Cube.Face4
Cube.Face5
Cube.Face6
```

to a new label `CubeSurface` one could use six CB cards, but with the wildcards this is much simpler to use just one CB card and specify the old label as

```
Cube.Face?
```

or also as

```
Cube.*
```

(depending on what other labels are also in the model).

Note that such wildcards are only supported in the *Old label* field of the CB card. The *New label* must be unique.

Another possibility to do a bulk renaming of labels is to read a label mapping table from an external file. This file consists of an abitrary number of lines

```
old_label   |  new_label
```

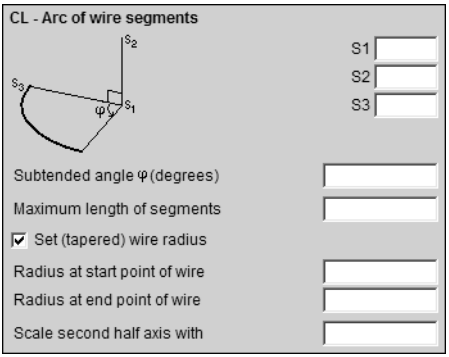
(i.e. the old and new label entries separated by the `|` character). Some external meshing programs can for instance export a NASTRAN file along with such a mapping table, and then by using the two commands

```
IN  3  3  "geometry.nas"
CB  "geometry.txt"
```

one can get the model into FEKO with the proper names of the parts (i.e. the file `geometry.txt` would then do a proper mapping of the NASTRAN property to the part name in the original CAD programme).

9.2.7 CL card

This card is used to create an arc consisting of wire segments.



Parameters:

S1: The centre point of the circle.

S2: A point perpendicular to the plane in which the circle lies and above its centre.

S3: The start point of the arc.

Subtended angle φ: The direction of the subtended angle, ϕ, is in the positive sense around the axis S1–S3. A negative value will create the arc in the opposite direction.

Maximum length of . . . : The maximum length of the segments that make up the arc. If this field is left empty, the value specified with the IP card is used. This length is in m and is scaled by the SF card.

Set wire radius: If checked, the radius set at the previous IP card is overridden for the current arc. This setting does not affect segments created after this card. Both radius values are in m and are affected by the SF card scaling factor. If only the start radius is specified, the arc will have a constant wire radius.

Radius at start . . . : The radius of the wire at the start point.

Radius at end . . . : The radius of the wire at the start point.

Scale second half axis: If this parameter is empty or is set to 1, a circular arc is created. If set to $\frac{b}{a}$, an elliptical arc is created. Here $\frac{b}{a}$ gives the ratio of the two half axes, where a is the distance S1–S3. It is recommended to generate elliptical arcs with extremely small or extremely large axial ratios with a CAD system as the distortion formulation used in PREFEKO may fail in these cases.

Quite often modelling the geometry of an arc requires shorter segments than those used for straight wires. Thus the maximum segment length specified with the IP card can be overridden along the arc by specifying a value in the field *Maximum length of segments*.

The radius of the arc is given by the distance between the points S1 and S3.

Examples of CL card usage:

The meshes shown in figures 9-10 and 9-11 are created with the CL card. The first figure shows the result of a wire arc with a uniform radius, and the second figure shows the result with an exaggerated taper specified.

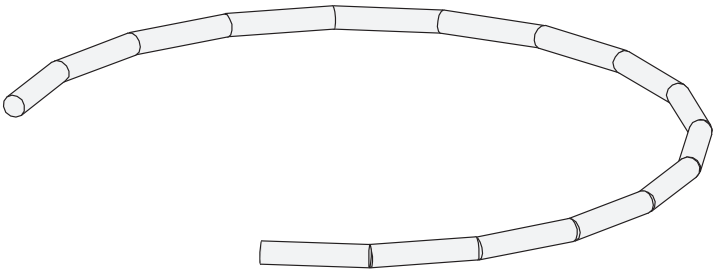


Figure 9-10: Example of a CL card from `demo_CL1.pre`

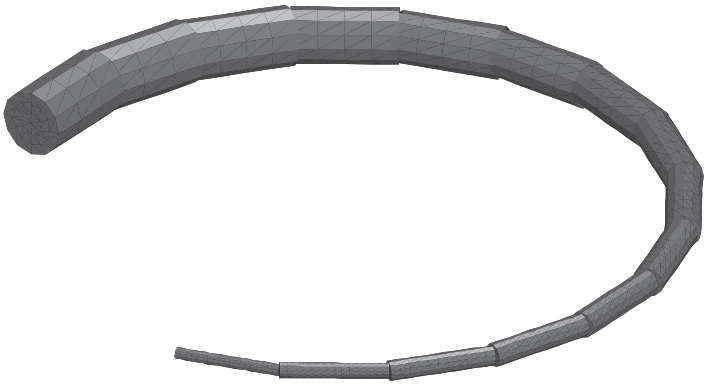
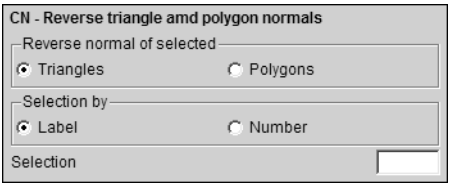


Figure 9-11: Example of a CL card with tapered wire radius from `demo_CL2.pre`

9.2.8 CN card

This card is used to reverse the normal direction of previously created triangles or polygons, for example after importing CAD data.



Parameters:

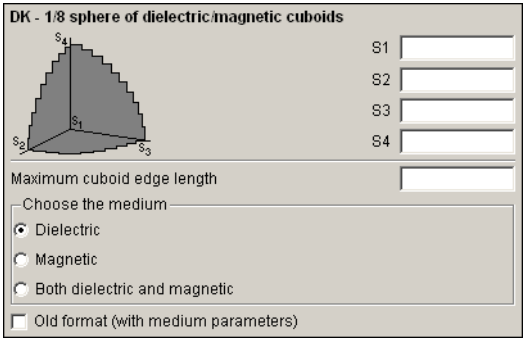
Reverse normal of . . . :	In this group, the user selects to reverse the normals of either *Triangles* or *Polygons*.
Selection by:	Here the user specifies whether the triangles / polygons are identified by their *Label* or absolute element *Number*.
Selection:	The label / element number of the triangles / polygons that must have their normals reversed.

The normal direction is important in some cases, such as when defining dielectric surfaces. For triangles, the normal vector is reversed by interchanging corners 1 and 3. For polygons the first point remains as is, but the corner points are listed in the opposite direction.

The CN card changes the normal of the affected triangles, but it does not change the settings of the ME card (which medium is on which side of the triangle as determined by the normal vector). For example, if the ME card is used to specify that the normal vectors of the triangles point from medium 5 to medium 2 then the application of the CN card will effectively change which medium lies on which physical side of the triangle.

9.2.9 DK card

This card is used to create an eighth of a sphere, meshed into cuboidal elements, for solutions using the volume equivalence principle in the MoM. The meshing parameters as set at the IP card are used, and the medium as set at the ME card is assigned to all created cuboidal elements.



Parameters:

S1: The centre of the sphere.

S2, S3, S4: Specify the three directions S1–S2, S1–S3 and S1–S4, that form the border of the eighth of the sphere. They must be perpendicular to each other and all three must have the same length (the sphere's radius).

Maximum cuboid edge length: The maximum side length of cuboids along the curved edge (in m) can be specified. This value is scaled by the SF card. If left empty, the value specified with the IP card is used.

Choose the medium: Select here whether the sphere is dielectric or magnetic or both (this is always with respect to the environment, e.g. if the relative permittivity ε_r of the cuboid material differs from the environment, then this is a dielectric sphere).

Old format (with medium parameters): For a detailed description of this parameter please see the QU card in section 9.2.34.

Dielectric bodies treated with the volume equivalence principle (using cuboids) cannot be used simultaneously with dielectric bodies treated with the surface equivalence principle or the FEM or with special Green's functions.

Example of DK card usage:

The DK card can be used to create a mesh of the eighth of a sphere as shown in figure 9-12.

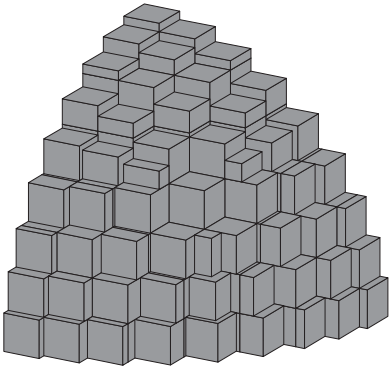
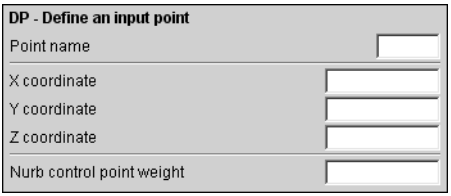


Figure 9-12: Example for the DK card from `demo_DK1.pre`

9.2.10 DP card

With this card points in space are defined. These points are used to define the extent and orientation of other geometric entities and to locate excitations.



To avoid ambiguity each point is assigned a name (a 5 character string). In the other cards (e.g. BL card) the points are referred to by their names.

Parameters:

Point name: Name (maximum of 5 characters) of the point.

X, Y, Z coordinate: Cartesian coordinates of the point in m (is scaled by the SF card).

Nurb control point weight: The weight of the control point when this point is used with the NU card (NURBS surfaces). If the field is empty, it defaults to 1.

In addition to its coordinates, each point is also assigned the current label (see LA card), so that a group of points can be selected by label, for example when moving points with the TP card.

Point names may use the characters `a-z`, `A-Z`, `0-9` and the special character `_` and no distinction is made between upper and lower case characters. Thus `P1a` and `p1A` refers to the same point.

In addition, when defining or using node names, simple variable names of the form `A#i` are allowed. The algorithm is that if a hash sign is found in a node point name, this hash sign and everything that follows is interpreted as a variable string, evaluated and rounded to the nearest integer. Thus if we have `#k=15` and use or define a point `P#k` then this is equivalent to using `P15` as point name. The length of the node name string (before and after expansion) is still limited to 5 characters.

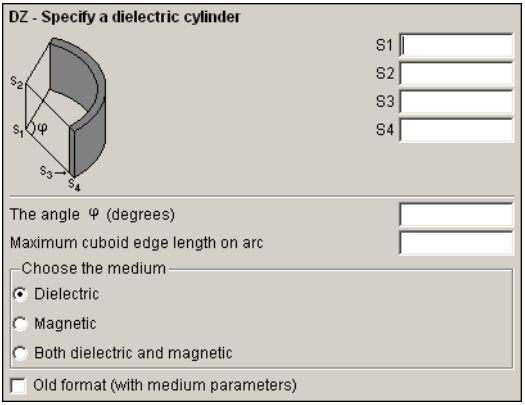
For instance, it would now be possible to define the points P1 to P20 inside a loop.

```
!!for #k = 1 to 20
DP   P#k           .....
!!next
```

Unlike most other geometry cards, the DP card (as well as the TP card) may also be used in the control section (after the EG card) of the `*.pre` file. This allows defining the points required by the AP card in this part of the file.

9.2.11 DZ card

This card is used to create a cylindrical shell, meshed into cuboidal elements, for solutions using the volume equivalence principle in the MoM. The meshing parameters as set at the IP card are used, and the medium as set at the ME card is assigned to all created cuboidal elements.



Parameters:

S1, S2: The start and end points, respectively, of the cylinder axis.

S3, S4: Points on the inside and outside, respectively, of the shell.

The angle φ: The angle of the cylindrical segment in degrees.

Maximum cuboid edge length on arc: Maximum edge length of the cuboids along the arc in m (is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.

Choose the medium: Select here whether the cylindrical shell is dielectric or magnetic or both (this is always with respect to the environment, e.g. if the relative permittivity ε_r of the cuboid material differs from the environment, then this is a dielectric cylinder).

Old format (with medium parameters): For a detailed description of this parameter please see the QU card in section 9.2.34.

Dielectric bodies treated with the volume equivalence principle (using cuboids) cannot be used simultaneously with dielectric bodies treated with the surface equivalence principle or the FEM or with special Green's functions.

Example of DZ card usage:

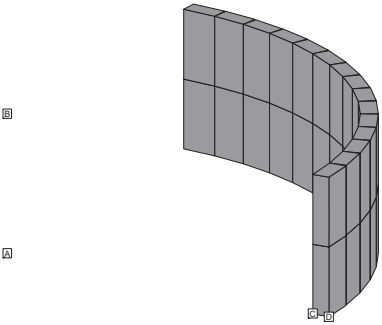
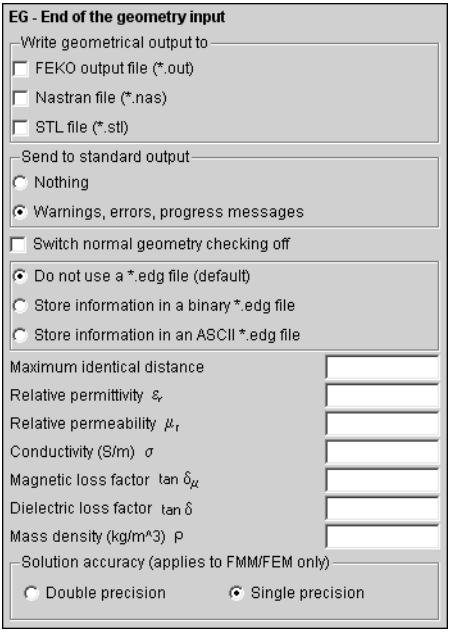


Figure 9-13: Example for the DZ card from `demo_DZ1.pre`

9.2.12 EG card

This card indicates the end of the geometrical input. It is essential that this card is used.


EG - End of the geometry input
Write geometrical output to
FEKO output file (*.out)
Nastran file (*.nas)
STL file (*.stl)
Send to standard output
Nothing
Warnings, errors, progress messages
Switch normal geometry checking off
Do not use a *.edg file (default)
Store information in a binary *.edg file
Store information in an ASCII *.edg file
Maximum identical distance
Relative permittivity ε_r
Relative permeability μ_r
Conductivity (S/m) σ
Magnetic loss factor $\tan\delta_\mu$
Dielectric loss factor $\tan\delta$
Mass density (kg/m^3) ρ
Solution accuracy (applies to FMM/FEM only)
Double precision
Single precision

Parameters:

Write geometrical output to: The geometry data of the segments and surface elements can be written to the FEKO output file, a Nastran format file, an STL file, or any combination thereof. The name of the Nastran or STL file will be the same as the FEKO model, but with a `*.nas` or `*.stl` extension. Writing the geometry data to the output file may lead to huge output files.

Send to standard output: If the field *Nothing* is selected, no messages are sent to the standard output device (usually the screen). If the item *Warnings, errors, progress messages* is selected, warnings, errors and messages that indicate the program's progress are sent to the standard output device.

Switch normal geometry checking off: If this item is checked, then verification of the geometry is switched off (see the discussion below this table).

**.edg options:* The `*.edg` file is used to store the connectivity information of the triangles. The file can be stored in binary or ASCII format. Binary format is used to keep the file size small. The `*.edg` file is read if it is present; and created — such that it may be read at a later run — if not. This is used to reduce the geometry set up computation time, especially for big models on large parallel computers. The ASCII formatted version of the `*.edg` file can be copied between platforms, (e.g. when preparing models on a PC and running FEKO on a workstation) but the file can become quite large.

Maximum identical distance: Two points less than this distance apart are considered identical. For an exact description see below. If this parameter is not specified, it is set internally to 10^{-6} m. In most cases this value is sensible. If scaling is done with the SF card, this value is also scaled. It is recommended to set this value, also referred to as EPSENT, only if FEKO demands it with a warning or error message.

Relative permittivity: The relative dielectric constant ε_r of the homogeneous medium in which all structures are contained. When this field is empty ε_r is set to 1.

Relative permeability: The relative permeability μ_r of the homogeneous surrounding medium. When this field is empty μ_r is set to 1.

Conductivity: The conductivity σ of the homogeneous surrounding medium in $\frac{1}{\Omega \mathrm{m}}$. When this field is empty σ is set to 0.

Magnetic loss factor: Magnetic loss tangent $\tan\delta_\mu$ of the homogeneous surrounding medium (the complex permeability is then given by $\mu = \mu_0\mu_r\left(1 - j\tan\delta_\mu\right)$).

Dielectric loss factor: Electrical loss tangent $\tan\delta$ of the homogeneous surrounding medium. This is an alternative way to specify the conductivity to which it is related by $\tan\delta = \frac{\sigma}{\omega\varepsilon_r\varepsilon_0}$.

Mass density: The medium density ρ in $\mathrm{kg/m^3}$ of the homogeneous surrounding medium. ρ is only required for the SAR computations and post processing of SAR results in POSTFEKO.

Solution accuracy: This parameter can be set to force FEKO to use single precision for the storage of some of the memory critical arrays for the MLFMM or FEM only. Single precision storage is the default behaviour for these methods.

The following should be noted regarding the export of the FEKO geometry to NASTRAN or STL:

- The STL export just dumps the data of all triangular patches of the FEKO model to an ASCII formatted STL file. Any other geometry (wires, tetrahedra etc.) is not exported since the STL format does not make provision for this. Also FEKO does not know any more any relation of the triangular patches to solids, and even in FEKO one can model just a plate for instance. Thus there is no special grouping of elements in the STL file, which does not represent a valid STL file in the strict sense. However, the exported information might still be useful in many cases.
- For the NASTRAN export, the wide column format is used to ensure that all significant digits are exported. Unlike the STL export, in NASTRAN all the various mesh elements used in FEKO are present. However, some information is lost, for instance for wire elements the thickness (wire radius) cannot be exported simply because the NASTRAN file format does not make provision for this. Also the NASTRAN property is used to represent the FEKO label. But since NASTRAN properties are just integer values and the FEKO label can be an arbitrary string, a mapping is done so that from each FEKO label just the associated number is used (see discussion in section 2.3) and exported then as the NASTRAN property.

The *Maximum identical distance* is used to set the tolerance in the mesh. The mesh information is created by the program PREFEKO, and stored in a `*.fek` file, in which all the triangles and segments are described by their corner points. Due to rounding errors it is possible that, for example, end points of connecting segments do not coincide. When searching for nodes, an ohmic connection is made when the difference is smaller than the *Maximum identical distance*.

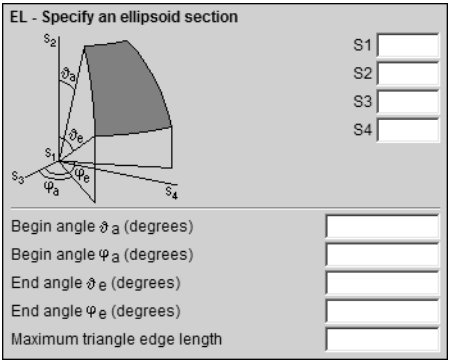
FEKO automatically checks for typical user errors that have been observed in the past. Examples of errors are connecting a wire segment to the middle of another wire, where the connection points do not coincide (see figure 2-2), or connecting surfaces that have different segmentation along the common edge (see figure 2-4). Such errors are detected if the parameter *Switch normal geometry checking off* is unchecked. The error detection routine should always be used. However, if the same geometry is to be used a number of times, the error detection can be disabled by checking this item.

If the surrounding medium is not vacuum, one can set the material parameters with the EG card as shown above. Alternatively the parameters of the surrounding medium can be set with the GF card which offers greater flexibility. For example, the GF card can be used to set the material parameters (as an arbitrary function of frequency) inside a frequency loop which is not possible with the EG card.

Note that switching to single precision currently does not have any effect on the MoM, PO, or UTD. These will always use double precision, irrespective of the switch. This might change in future. Presently, switching to single precision only has an effect for the MLFMM (multilevel fast multipole method) and the FEM (finite element) solution.

9.2.13 EL card

A mesh of surface triangles in the shape of an ellipsoidal section can be created with this card.



Parameters:

S1: The centre point of the ellipsoid.

S2: A point, in the direction $\vartheta = 0°$ in elliptical coordinates. The distance of the two points S1 and S2 determines the half-axis of the ellipsoid in this direction.

S3: A point in the direction $\vartheta = 90°$, $\varphi = 0°$ in elliptical coordinates. The distance of the two points S1 and S3 determines half of the axis of the ellipsoid in this direction.

S4: A point in the direction of the third coordinate, i.e. the axes S4–S1, S3–S1 and S2–S1 must be perpendicular. The distance of the two points S1 and S4 determines half of the axis of the ellipsoid in this direction.

Begin angle ϑ_a: Start angle of the ellipsoid in degrees.

Begin angle φ_a: Start angle of the ellipsoid in degrees.

End angle ϑ_e: End angle of the ellipsoid in degrees.

End angle φ_e: End angle of the ellipsoid in degrees.

Maximum triangle edge length: Maximum length of the triangles along the curved edge in m (is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.

Note that the angles ϑ and φ are defined in an elliptical, rather than in a spherical coordinate system. For a Cartesian coordinate system with origin S1, x axis in direction of S3, y axis in the direction of S4 and z axis in the direction of S2, a point on the surface of the ellipsoid is given as

$$\vec{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a \sin\vartheta \cos\varphi \\ b \sin\vartheta \sin\varphi \\ c \cos\vartheta \end{bmatrix}$$

where the lengths a, b and c are the lengths of the ellipsoid's three half-axes. (For example the length a is the distance between the points S3 and S1).

The normal vector of the generated triangles always points outwards. The algorithm used for the segmentation can fail if the ratio of the half-axis is too extreme, for example if the longest half-axis is a factor 100 longer than the shortest. It is strongly advised to check the geometry with POSTFEKO.

Example of EL card usage:

The mesh shown in figure 9-14 is generated by using the EL card.

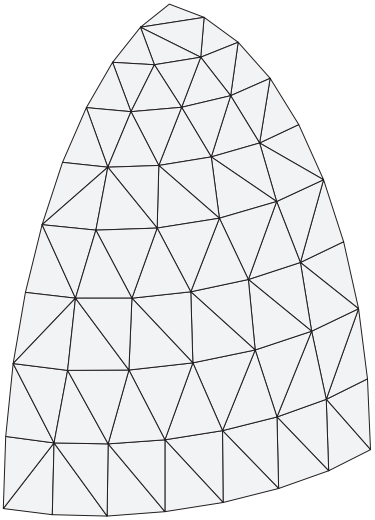


Figure 9-14: Example for the EL card from `demo_EL.pre`

9.2.14 FM card

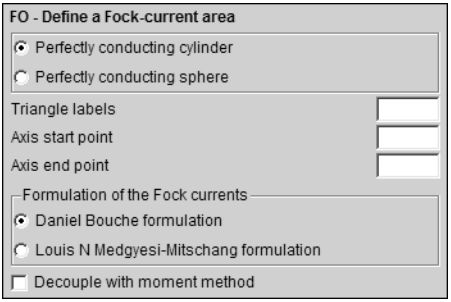
This card is used to instruct the FEKO solver to calculate the solution using the Multilevel Fast Multipole Method (MLFMM) instead of the MoM on all structures in the simulation.

Parameters:

Box size at fines level: The MLFMM is based on a hierarchical tree-based grouping algorithm, and depending on the frequency and the model dimensions FEKO automatically determines the number of levels in this tree and the size of the boxes at the finest level. It is also recommended that this default box size of $0.23\,\lambda$ is kept. When there is no convergence in the MLFMM, then advanced users might try to slightly increase or decrease this box size by setting it manually (the input is in terms of the wavelength).

9.2.15 FO card

With this card an area is defined in which the surface current density is an approximation according to the Fock theory.



Parameters:

Perfectly conducting cylinder: Select this option if the Fock area is/resembles a cylinder.

Perfectly conducting sphere: Select this option if the Fock area is/resembles a sphere.

Triangle labels: The label of the metallic triangles that form the surface of the Fock area (e.g. the surface of the cylinder).

Axis start / Axis end point: These dialogs are only visible for Fock cylinders and should correspond to the start and end points of the cylinder axis.

Sphere centre point: This dialog is only visible for Fock spheres and should correspond to the centre of the sphere.

Formulation of the Fock currents: The type of process for the Fock currents, either the Method by Daniel Bouche or the Method by Louis N. Medgyesi–Mitschang.

Decouple with moment method: Check this item to force FEKO to neglect the coupling between the MoM and Fock regions, so that there is no feedback by which the Fock currents may influence the current distribution in the MoM region. This option, which is particularly applicable when the MoM and Fock regions are not in close proximity, should result in a considerable reduction in computational effort and storage space.

The radius of the cylinder or sphere does not have to be defined. It is determined by the distance to the metallic triangles, with the label specified in *Triangle labels*.

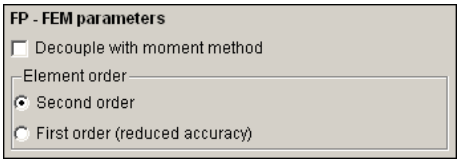
The cylinder Fock currents can also be applied to cones (KK card, approximated by a staircase construction of cylinders) and sections of a torus that resembles a cylinder

(TO card). Although the FO card is strictly only applicable to spherical and cylindrical surfaces, it is often a good approximation on conical and toroidal surfaces.

It must be noted that the search for creeping rays on the Fock surface does not take into account multiple Fock regions, i.e. one creeping ray can only exist in one Fock region. Therefore when for instance modelling a sphere and using symmetry, it is highly advisable to create part of the sphere, then use the SY card to mirror it, and only then use one FO card which applies to the whole sphere. When using SY cards or also TG cards, they do operate on already existing Fock regions defined above these cards and mirror them also or move them, but with SY the problem just is that after the symmetry there exist then multiple Fock regions, and the creeping rays along geodesic lines stop at the boundary of each Fock region.

9.2.16 FP card

With this card options related to the FEM (*Finite Element Method*) can be set.



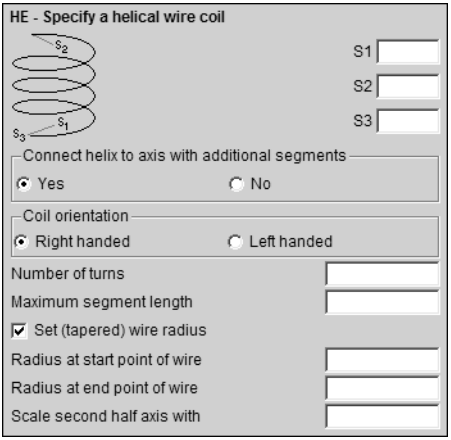
Parameters:

Decouple with moment method: Normally within the MoM/FEM hybrid method these two techniques are fully coupled, and also the surface of the FEM region is treated by the MoM. For certain applications when there is a larger separation between the MoM and the FEM regions (e.g. human body with a GSM base station antenna) this decoupling checkbox can be checked. Then similar to switching off the coupling for the MoM/PO or MoM/UTD hybrid methods, first the MoM region is solved for by neglecting the FEM domain, and this MoM solution is then used as impressed excitation for the FEM. Also the MoM is in this de-coupled case no longer used on the FEM surface, but rather an absorbing boundary condition is applied there. The advantage of this decoupling is a saving of memory and computation time.

Element order: When second order is selected, then FEKO uses hierarchal tetrahedral elements with LT/QN (linear tangential/quadratic normal) vector basis functions for the electric field inside the FEM region. On the boundary surface of the FEM region CT/LN (constant tangential/linear normal) vector basis functions are used for the equivalent electric and magnetic surface currents. When first order is selected, then CT/LN basis functions are used everywhere, on the boundary and inside. When switching to first order normally a finer mesh is required to get the same solution accuracy than second order. Thus second order is the default. However, when having a fine mesh (like modelling details of a biological structure), then one might consider switching to first order only to reduce the number of unknowns.

9.2.17 HE card

With this card a helical coil, consisting of wire segments can be created.



Parameters:

S1: The start point of the coil's axis.

S2: The end point of the coil's axis.

S3: The start point of the windings.

Connect helix . . . : Create connections from the two ends of the coil to the axis (at points S1 and S2). See also left side of figure 9-15. If the connections are not generated, point S3 is a connection point. See also the right side of figure 9-15.

Coil orientation: Indicate whether a right- or left handed coil should be created.

Number of turns: In this field the number of turns for the helix is entered. It need not be an integer number.

Maximum segment length: Maximum length of the segments, that are used for the windings in m (is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.

Set (tapered) radius . . . : If this item is checked, a tapered wire radius can be set. Normally the wire radius is set with the IP card. Checking this item overrides this radius for the current helix without affecting the default for later segments (The radius is in m and is affected by the SF card scaling factor.) The segments connecting to the axis are not tapered and have radii corresponding to the start point and end point respectively.

Radius at start . . . :	The radius of the wire at the start of the coil.
Radius at end . . . :	The radius of the wire at the end point of the coil.
Scale second half axis:	If this parameter is empty or is set to 1, a helix with a circular cross section is created. If set to $\frac{b}{a}$, a helix with an elliptical cross section is created. Here $\frac{b}{a}$ gives the ratio of the two half axes, where a is the distance S1–S3. It is recommended to generate elliptical helices with extremely small or extremely large axial ratios with a CAD system as the distortion formulation used in PREFEKO may fail in these cases.

Quite often modelling the geometry of the coil requires shorter segments than those used for straight wires. Thus the maximum segment length specified by the IP card can be overridden along the arc by setting *Maximum segment length*.

The windings are generated between the two points S1 and S2, that lie on the axis. The radius of the coil is defined by the distance between the points S1 and S3. For elliptical cross sections this is the length of one half axis and the other one is *Scale second half axis* times this length.

Example of HE card usage:

The two coils shown in figure 9-15 are created using the HE card.

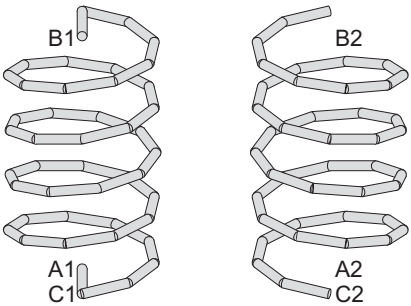


Figure 9-15: HE card examples — the coil on the right is coiled in the left handed direction. These examples are found in `demo_HE1.pre`

9.2.18 IN card

This card is used to include external files. These files may be other `*.pre` files (which are included as if they were part of the master file) *or* mesh data files containing wire segments, triangles, quadrangles, tetrahedral volume elements and/or polygonal plates (in FEMAP neutral, an ASCII format, NASTRAN, meshed AutoCAD `*.dxf`, NEC model, Concept model, STL, PATRAN neutral or ANSYS `*.cdb` files).

Some fields are common to more than one option:

File name: The name of the file. This parameter is required for all import options. The filename may contain directory names as well, for example, `..\myfiles\include.inc` and will have different extensions for the various import options. Both \ and / are allowed on Windows and UNIX systems.

Include segments: Check this item to include all wire segments that match the label selection. (See below.)

Include triangles: Check this item to include surface triangles.

Include quadrangles: Check this item to include quadrangles. The quadrangles are subdivided into triangles (along the shortest diagonal) during importation.

Include polygons: Check this item to include polygonal plates (for UTD).

Include node points: Check this item to include node points.

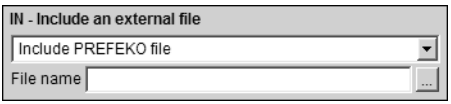
Include only node points for imported triangles and/or wires: If this item is checked, only the node points which are used by the imported elements, are imported. This is useful if one imports, for example, a few segments from a file containing a large number of triangles. With this option one may then only import the points associated with the segments — even if they have the same label as the ones associated with triangles only.

Label selection: Most options allow label selective importing. (How the various layers / properties / names are converted to FEKO labels is discussed separately for each import option.) One may *Include all items, Include items with only a single label* or *Include items with range of labels*. If the first option is selected all elements are imported, irrespective of label. If the single label option is selected, the *Include structures with...* field becomes active. Specify the label (as it will be after conversion) in this field. If the range option is selected, the *Up to...* field also becomes active. All elements with the label larger or equal to the first and smaller or equal to the second field, are included. If the import option does not support label selection, all elements are imported.

Scale factor: An optional constant scaling factor can be applied to the imported geometry. This is necessary, for example, if separate CAD files with different units must be imported, or if the `*.pre` is, for example, created using mm while the CAD file is constructed using inches as unit. It should be noted that the scaling factor specified here is applied in addition to any scaling factor that may be set with the SF or TG cards.

9.2.18.1 PREFEKO file

This option is used to include cards and commands (such as variable definitions etc.) from a separate file. One may, for example, create a single file with an antenna which is then imported into different environment models.



For this option, the file name is the only parameter. This file is included as if it was part of the master `*.pre` file. These include files usually have the extension `*.inc`, but can have any extension. The cards and instructions in the included files are processed as if they were part of the main file. Therefore points and labels defined in the included file remain valid in the remainder of the main file. Note that it is also possible to use such an IN card in the control section of the `*.pre` file (for instance to import some feed model).

When reading a PREFEKO file it is not possible to add a scaling factor to the IN card. In this case the TG card must be used if the global (SF card) scaling option is not sufficient.

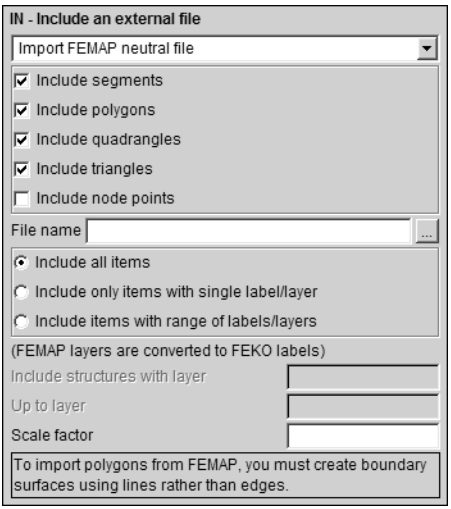
It is possible to use multiple nested levels of include files (i.e. one include file can include another one and so on). It is also possible to specify together with the filename an absolute or relative path like in

```
IN   0  "..\subdir\file.inc"
```

In such a case — if multiple levels of include files are used — it is first tried to find the include file using the path relative to the location of the file where the IN card is used. If the include file is not found there, then PREFEKO also tries to find the include file using the path relative to the location of the main `*.pre` file which is processed.

9.2.18.2 FEMAP neutral file

This option is used to import models generated by the commercial CAD meshing program FEMAP. The models must be exported from FEMAP in the `*.neu` FEMAP neutral file format.


IN - Include an external file
Import FEMAP neutral file
Include segments
Include polygons
Include quadrangles
Include triangles
Include node points
File name
Include all items
Include only items with single label/layer
Include items with range of labels/layers
(FEMAP layers are converted to FEKO labels)
Include structures with layer
Up to layer
Scale factor
To import polygons from FEMAP, you must create boundary surfaces using lines rather than edges.

This card supports all the parameters described in the general section of the IN card above.

The label selection uses the FEMAP layer numbers which are converted to FEKO labels. Wires must be meshed into elements which are imported as segments, surfaces into triangles or quadrangles which are imported as FEKO triangles, and boundary surfaces are imported as polygons. The boundary surface must be bordered with *line* curves rather than *edge* curves.

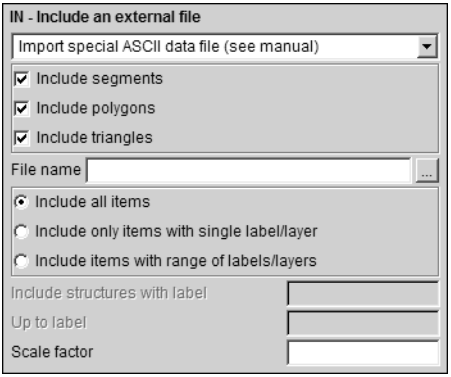
The user can also elect to import points from the `*.neu` file. All points defined as such in FEMAP are then available in PREFEKO as points (as if they were defined by DP cards) of the form `Pxxx` where `xxx` is the point ID in FEMAP. This may be used, for example, when attaching additional structures to a geometry partly created in FEMAP. In addition, the coordinate values of the point are available as variables in PREFEKO. For example, the variables `#p1234x`, `#p1234y` and `#p1234z` give the coordinates of the FEMAP point with ID 1234. Note that points are not included by default.

It should be remembered that it is not possible to specify a wire radius in FEMAP. Thus the wire radius must be specified by an IP card preceding the IN card. Similarly, when specifying the surface of a dielectric, the IN card must be preceded with the correct ME card (completely analogous to the case without FEMAP).

POSTFEKO should be used to verify the included geometry.

9.2.18.3 Import special ASCII data file

This option is used to import meshes stored in the geometry data file as specified below.



The ASCII format only supports segments, triangles and polygons, but all other (non-selection) parameters discussed in the general section of the IN card above apply. In this case the label is specified directly in the file and no conversion is required.

Dielectric triangles or metallic triangles which form the surface of a dielectric, are created by preceding the IN card with the appropriate ME card. (In exactly the same way as is the case without the IN card.)

The data of the segments, triangles and polygonal plates are given in an ASCII file, formatted as shown below. There is no need to adhere to specific columns, the data fields merely have to be separated by one or more spaces.

```
nk   nd   ns   np   nt
x(1)   y(1)   z(1)  (String_name)
x(2)   y(2)   z(2)  (String_name)
...

x(nk)  y(nk)  z(nk)  (String_name)
d1(1)  d2(1)  d3(1)  0  (Label)
d1(2)  d2(2)  d3(2)  0  (Label)
...

d1(nd) d2(nd) d3(nd) 0  (Label)
s1(1)  s2(1)  0      0  (Label)
s1(2)  s2(2)  0      0  (Label)
...

s1(ns) s2(ns) 0      0  (Label)
```

```
nnp(1)   p1(1)  p2(1)  p3(1)  ...   (Label)
nnp(2)   p1(2)  p2(2)  p3(2)  ...   (Label)
...

nnp(np)  p1(np) p2(np) p3(np) ...   (Label)
```

The meaning of the above is:

nk	Number of nodes
nd	Number of triangles
ns	Number of segments
np	Number of polygonal plates
nt	Number of tetrahedral volume elements (defaults to 0 if not specified)
$x(i)$	x coordinates of node i in m (is scaled by the SF card)
$y(i)$	y coordinates of node i in m (is scaled by the SF card)
$z(i)$	z coordinates of node i in m (is scaled by the SF card)
$d1(j)$	Number (index) of the first vertex of triangle j
$d2(j)$	Number (index) of the second vertex of triangle j
$d3(j)$	Number (index) of the third vertex of triangle j
$s1(k)$	Number (index) of the starting point of segment k
$s2(k)$	Number (index) of the end point of segment k
$nnp(m)$	Number of corner points in polygon m
$p1(m)$	Number (index) of the first corner of polygon m
$p2(m)$	Number (index) of the second corner of polygon m
$p3(m)$	Number (index) of the third corner of polygon m
...	Number (index) of the additional corners of polygon m
String_name	Optional string name of the point. It must be a string of up to 5-characters, similar to the point name of the DP card. If a point is named, it can be used in any card following the IN card.
Label	Specifying the label as the last parameter of any structure is optional. If no label is specified, the value defined at the last LA card will be used. Note that if a label or range of labels is specified (with parameters after the filename), this LA card label will be used to determine if a structure is included or not.

The radius of segments must be specified by an IP card before the IN card. It is recommended to check the geometry with POSTFEKO.

Example:

The structure in figure 9-16, consisting of 5 node points and 3 triangles with label 7 (no segments or polygonal plates), may be imported from the following data file

```
5  3  0  0
3.0  0.0  1.0
4.0  2.0  1.0
2.5  3.0  2.5
0.0  3.0  4.0
1.0  0.0  3.0
1  2  3  0  7
1  3  5  0  7
3  4  5  0  7
```

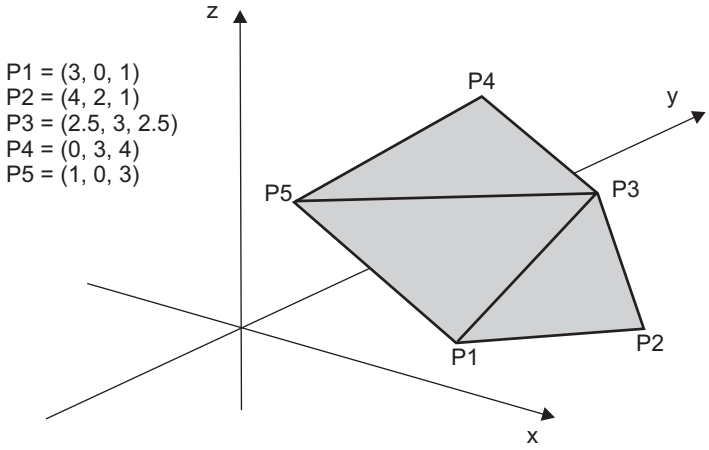
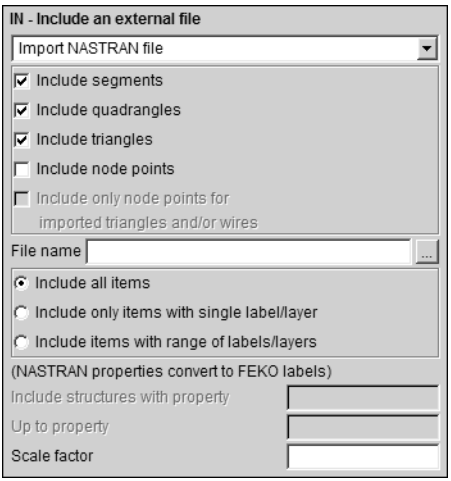


Figure 9-16: Example for IN card

9.2.18.4 Import NASTRAN file

With this option, PREFEKO can import a model from a NASTRAN file. It supports both 8-character and 16-character wide column based files and comma separated files. Only the keywords GRID, CTRIA3, CQUAD4, CBAR and CROD for nodes, triangles, quadrangles (divided into two triangles along the shortest diagonal) and segments are processed.



NASTRAN does not support polygonal plates, but all other parameters in the general section of the IN card above apply. The label selection uses the NASTRAN properties which are converted to FEKO labels.

As when importing FEMAP neutral files, the wire radius must be set with the IP card preceding the IN card, and an ME card must be used when specifying dielectric surfaces in the same way as when the IN card is not present.

Since grid points do not have an associated property, points are imported irrespective of their label, but they may be limited to those used for the imported geometry.

The user can also import points from the NASTRAN file. The points defined in the NASTRAN file will then be available in PREFEKO as points (as if they were defined by DP cards) of the form `Nxxx` where `xxx` is the index of the grid point. This may be used, for example, to attach additional structures to the geometry. In addition, the coordinate values of the point are available as variables in PREFEKO. For example, the variables `#n1234x`, `#n1234y` and `#n1234z` give the coordinates of the NASTRAN grid point with index 1234. Note that points are not included by default.

Each line in the 8-character column based format consists of one keyword such as "GRID" starting in column 1. From column 9 onwards follow 9 input fields with widths of 8 characters each. Thus input field 1 uses columns 9 to 16, input field 2 uses columns 17 to 24 etc. The ninth (and last) input field 9 ends at column 80. Below is a very simple NASTRAN example file consisting of a plate (property 1; subdivided into eight triangles) and a rod (property 2; subdivided into two segments).

```
|1       |9       |17      |25      |33      |41      |49      |49
ID XXXXXXXX,YYYYYYYY
CEND
BEGIN BULK
GRID            1              0.0     0.0     0.0
GRID            2          0.50000     0.0     0.0
GRID            3          1.00000     0.0     0.0
GRID            4              0.0 0.50000     0.0
GRID            5          0.50000 0.50000     0.0
GRID            6          1.00000 0.50000     0.0
GRID            7              0.0 1.00000     0.0
GRID            8          0.50000 1.00000     0.0
GRID            9          1.00000 1.00000     0.0
GRID           10          0.50000 0.50000 2.00000
GRID           11          0.50000 0.50000 1.00000
CROD            9       2       5      11
CROD           10       2      11      10
CTRIA3          1       1       4       5       8
CTRIA3          2       1       4       8       7
CTRIA3          3       1       5       6       9
CTRIA3          4       1       5       9       8
CTRIA3          5       1       1       2       5
CTRIA3          6       1       1       5       4
CTRIA3          7       1       2       3       6
CTRIA3          8       1       2       6       5
ENDDATA
```

For the node points FEKO also supports 16 character wide input fields. The keyword `GRID` in columns 1 to 4 is followed by a star and three spaces. The node ID is then in columns 9 to 24, the x coordinate in columns 43 to 56, y in columns 57 to 72 and z in columns 9 to 24 of the next line.

For example

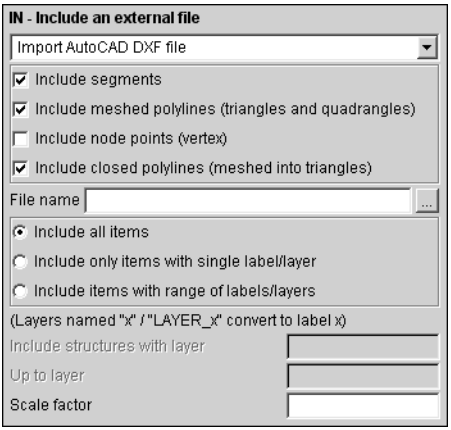
```
|1       |9                  |25    ...   |43              |57                |73       |81
GRID*                     1                50.000000000   -18.480176926       1
*       1    23.222875595
GRID*                     2                50.000000000   -18.480176926       2
*       2   -13.410394669
```

For the comma separated format, the individual entries are separated by commas

```
GRID,1,0,-238.533,186.7983,0.000000,0
GRID,2,0,-244.777,214.3057,172.9991,0
GRID,3,0,288.0060,115.1831,339.8281,0
GRID,4,0,356.2201,50.15516,0.000000,0
CTRIA3,1,1,1,2,3,,0.0,,
CTRIA3,2,1,1,2,4,,0.0,,
```

9.2.18.5 Import AutoCAD DXF file

This card allows importing `*.dxf` models. The `*.dxf` file must comply with the *release 12* DXF format specifications. It should contain *meshed* or *closed* polylines surfaces (see the discussion below) and lines (that will be segmented by PREFEKO as discussed below).


IN - Include an external file
Import AutoCAD DXF file
Include segments
Include meshed polylines (triangles and quadrangles)
Include node points (vertex)
Include closed polylines (meshed into triangles)
File name
Include all items
Include only items with single label/layer
Include items with range of labels/layers
(Layers named "x" / "LAYER_x" convert to label x)
Include structures with layer
Up to layer
Scale factor

In addition to the file name, label selection and scale factor discussed in the general section of the IN card above, the DXF import option supports the following element selection:

Include segments: Same as above.

Include meshed polylines (triangles and quadrangles): Check this item to include meshed polylines (triangles and quadrangles) from the DXF file into the model.

Include node points: Check this item to include node points from the DXF file into the model.

Include closed polylines (meshed into triangles): Check this item to include closed polylines from the DXF file into the model. These structures will be meshed by PREFEKO.

Layers named `n` or `LAYER_n` (where `n` is an integer number) in the `*.dxf` model are converted to label `n` in FEKO. For all structures for which no label is defined in this format, the label specified with the last LA card preceding the IN card is used. (If no such LA card is in effect, the default is label 0.) This label is used in the label selection.

As for the other meshed CAD formats, dielectric triangles or metallic triangles which form the surface of a dielectric, are created by preceding the IN card with the appropriate ME card.

PREFEKO only processes the geometry information in the section of the file between the keywords `ENTITIES` and `ENDSEC`.

The user can import points (i.e. vertices of polylines and start/end points of lines) from the DXF file. The points defined in the DXF file will then be available in PREFEKO as points of the form `Qxxx` where `xxx` is a consecutive point index. In addition, the coordinate values of the point are available as variables in PREFEKO. For example, the variables `#q1234x`, `#q1234y` and `#q1234z` give the coordinates of point 1234. Note that points are not included by default. The imported node points will have the label (i.e. converted layer) of the corresponding LINE or POLYLINE structure (the layer of the VERTEX block for polylines is not used). A label range selection at the IN card may be applied such that only the points with a correct layer will be imported.

Segments are imported from blocks defined by the keyword `LINE`.

```
  0
LINE
  8
LAYER_01
  .
  .
 10
-0.0538
 20
0.0
 30
8.134
 11
5.110
 21
2.857
 31
0.0
  0
 ... (next keyword)
```

The group code `8` at some point below `LINE` indicates that the next line contains the layer name. In this case, the layer will be converted to label 1. The line will be imported and segmented if this label lies in the required range. (If not, PREFEKO will search for the next occurrence of `LINE`.) Next the x, y and z components of the start point follow the group codes `10`, `20` and `30`; and those of the end point follow the codes `11`, `21` and `31`.

Here the start and end points are (x, y, z) = (-0.0538, 0.0, 8.134) and (5.110, 2.857, 0.0) respectively. If any of the coordinate group codes are absent (such as in a 2D model), the related coordinate is set to zero. The block is terminated by the group code `0`. The wire is segmented according to the maximum segment length specified by the IP card, and the segments also have the radius specified by this card.

Meshed surfaces are imported from blocks denoted with the keyword `POLYLINE`. This block contains the layer name (following the group code `8` as before; if there is no group code `8` before the first `VERTEX`, the label specified with the last LA card will be used) and a number of `VERTEX` structures. There can be an arbitrary number of `VERTEX` structures, but there should be at least four.

The `POLYLINE` structure is terminated by the keyword `SEQEND`.

```
  0
POLYLINE
  8
LAYER_02
  .
  .
VERTEX
  .
  .
VERTEX
  .
  .
VERTEX
  .
  .
  0
SEQEND
```

There are two types of vertices. The first type defines points in space

```
  0
VERTEX
  8
LAYER_02
  .
  .
 10
7.919192
 20
3.393939
 30
0.0
  .
  .
  0
 ... (next keyword)
```

where the x, y and z components of the point follow the group codes `10`, `20` and `30`. The layer information is ignored.

The second type of vertex is a "linker".

```
  0
VERTEX
  8
LAYER_02
  .
  .
 70
   128
 71
    4
 72
    2
 73
    1
 74
    3
  .
  .
  0
 ... (next keyword)
```

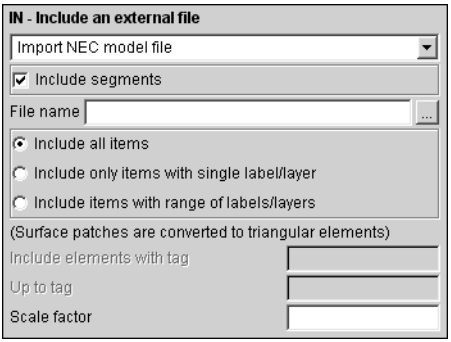
which defines a triangle or quadrangle by specifying the indices (starting from 1 in the order the non-linker vertices are specified) of the vertices which form its corner points. Vertices are defined as linkers by setting a value of `128` in the group code `70` field. For linker vertices the coordinates are ignored. Note that some old `*.dxf` versions do not contain linker vertices — they cannot be imported. (Usually they do not contain mesh information.)

The four integer numbers after the group codes `71`, `72`, `73` and `74` give the indices of corners of the triangle or quadrangle. (In the case of a triangle one of these is absent.) PREFEKO divides each quadrangle into two triangles along the shortest diagonal.

In addition to being able to import meshed polylines, closed polylines can also be imported. These will be meshed into triangular patches during the import according to the meshing parameters set at the IP card.

9.2.18.6 Import NEC model file

PREFEKO also supports importing wire geometry from NEC[4] models. Note that NEC models usually consist of wire grid surfaces and it would be more efficient to convert the models to FEKO surfaces, but this cannot be done automatically.



For this option only the file name, label selection and *Include segments* are supported.

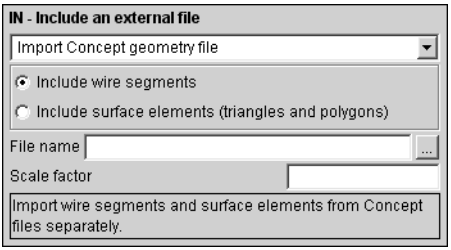
The label selection uses the NEC tags which are converted to FEKO labels. This applies to the tag when the element is defined. If the tag is modified after the inclusion (for example with the GM card) the elements with the modified tag are also included. The type selection parameter `x` is also supported, but it may only have the value `1` for wire segments.

The NEC import filter considers only the geometry cards CM, CE, GA, GW, GM, GR, GS, GX and GE. A warning is given if other cards are encountered. If the model contains multiple geometries only the first one is read.

[4] G.J. Burke and A.J. Poggio, "Numerical Electromagnetics Code (NEC) — Method of Moments," Lawrence Livermore Laboratory, January 1981.

9.2.18.7 Import Concept geometry file

With this option one may import CONCEPT[5] geometry files.


IN - Include an external file
Import Concept geometry file
Include wire segments
Include surface elements (triangles and polygons)
File name
Scale factor
Import wire segments and surface elements from Concept files separately.

Since CONCEPT uses two different files for wires and surface elements, the type of element selection is obligatory and determines the type of geometry file to be read. The only other parameters supported here are the file name and scale factor. (The CONCEPT file does not contain labels.) If the concept model contains quadrangles, they are divided into triangles.

Since wires don't have a radius in the model files, the radius is specified with a preceding IP card. Likewise, the elements don't have labels, and the label as specified at the last LA card before the IN card is used. If there is no LA card, the label defaults to zero.

As for the CAD models, dielectric triangles or metallic triangles which form the surface of a dielectric, are created by preceding the IN card with the appropriate ME card.

The CONCEPT files for wires are as follows

```
number_of_wires
x_start y_start z_start x_end y_end z_end   [number_of_wires times]
```

where the first integer specifies the number of wires followed by the coordinates of the start and end point of each wire. The file is completely free format — the values are just separated by white space. The surface file is

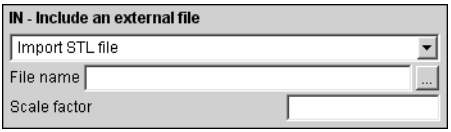
```
number_of_nodes   number_of_patches
x y z                                       [number_of_nodes times]
p1 p2 p3 p4                                 [number_of_patches times]
```

again using free format. The values `x`, `y` and `z` specify the node coordinates and `p1`, `p2`, `p3` and `p4` specify the corner nodes of the triangles (in this case `p4` is `0`) and quadrangles.

[5] A MoM code developed at the University of Hamburg–Harburg, Germany.

9.2.18.8 Import STL file

PREFEKO can also import STL — both ASCII and binary — files. STL files supports only triangular patches and these are all imported. Also, since the STL file makes no provision for any labels, label selection is not supported. The scale factor is supported.



An example of an ASCII STL file is

```
SOLID CATIA STL PRODUCT
  FACET NORMAL -4.602166E-01 -1.858978E-01 -8.681260E-01
    OUTER LOOP
      VERTEX    4.789964E-01 -8.440244E-01  2.878882E-01
      VERTEX    4.764872E-01 -8.439470E-01  2.892018E-01
      VERTEX    4.783065E-01 -8.414296E-01  2.876983E-01
    ENDLOOP
  ENDFACET
  FACET NORMAL -4.601843E-01 -1.859276E-01 -8.681367E-01
    OUTER LOOP
      VERTEX    4.764872E-01 -8.439470E-01  2.892018E-01
      VERTEX    4.761175E-01 -8.425569E-01  2.891001E-01
      VERTEX    4.783065E-01 -8.414296E-01  2.876983E-01
    ENDLOOP
  ENDFACET
ENDSOLID
```

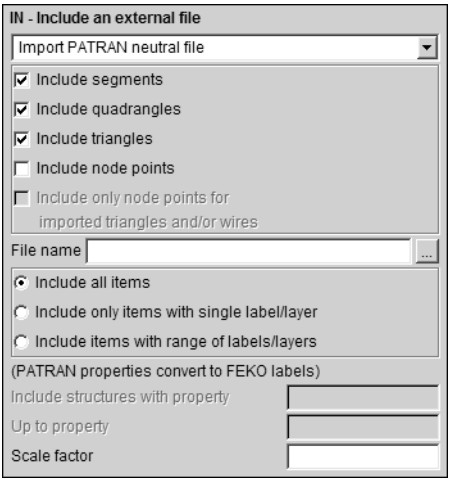
For the description of binary STL files, please see
http://www.ennex.com/fabbers/StL.asp
http://rpdrc.ic.polyu.edu.hk/old_files/stl_binary_format.htm

9.2.18.9 Import CADFEKO model file

CADFEKO exports the mesh to a CFM file which is imported by an IN card in the default PRE file created by CADFEKO. The options are similar to those of the other formats that PREFEKO can import.

9.2.18.10 Import PATRAN neutral file

PREFEKO also supports importing PATRAN files.



PATRAN does not support polygonal plates, but all other parameters in the general section of the IN card above apply. The label selection uses the PATRAN material ID's which are converted to FEKO labels. Only the following PATRAN neutral packet types are imported:

- `01` Node data (all coordinates are interpreted in the global rectangular frame, local coordinate frames are not supported).
- `02` Element data. The shapes 2 = bar, 3 = tri, 4 = quad and 5 = tet are allowed. Quadrangles are automatically subdivided into triangles along the shortest diagonal.
- `99` End of file flag.

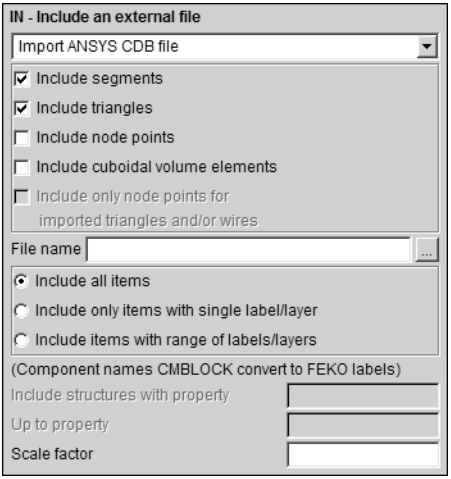
Other packet types are ignored.

As when importing `*.neu` files, the wire radius must be set with the IP card preceding the IN card, and an ME card must be used when specifying dielectric surfaces in the same way as when the IN card is not present.

The user can also import points from the PATRAN file similar to importing points from FEMAP or NASTRAN files. The points defined in the PATRAN file will then available in PREFEKO as points (as if they were defined by DP cards) of the form `Txxx` where `xxx` is the index of the grid point. This may be used, for example, to attach additional structures to the geometry. In addition, the coordinate values of the point are available as variables in PREFEKO. For example, the variables `#t1234x`, `#t1234y` and `#t1234z` are set to the coordinates of the point with index 1234. Note that points are not included by default. Since points do not have an associated property ID, points are imported irrespective of their label.

9.2.18.11 Import ANSYS CDB file

PREFEKO also supports importing geometry from ANSYS `*.cdb` files. By default, when exporting such files from ANSYS, the BLOCKED option is used. PREFEKO only understands this BLOCKED syntax, the UNBLOCKED version is not supported. Also regarding the element type, only the ANSYS element types 200 (filaments, triangles, tetrahedral elements, and brick elements) as well as element type 16 (`pipe16`, wire with a finite radius) are supported.


IN - Include an external file
Import ANSYS CDB file
Include segments
Include triangles
Include node points
Include cuboidal volume elements
Include only node points for imported triangles and/or wires
File name
Include all items
Include only items with single label/layer
Include items with range of labels/layers
(Component names CMBLOCK convert to FEKO labels)
Include structures with property
Up to property
Scale factor

The selection of polygons and quadrangles are not supported, but all other (non-selection) options discussed in the general section of the IN card is supported. It also supports an additional selection option:

Include cuboidal volume elements: Check this item to include cuboidal elements to be used with the volume equivalence principle in FEKO.

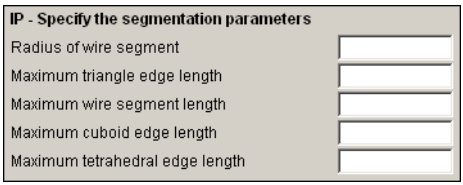
The component name from the CMBLOCK is converted to the FEKO label. Since the FEKO label must be an integer value, only component names which are integer strings (for example, `15`) or end with an underscore followed by an integer string (for example, `FEED_7`) will be converted to FEKO labels (15 and 7 in the examples above). In all other cases (for example, for a component name `PATCH`) the FEKO label will be set to zero.

Note unlike most of the other CAD import formats supported by the IN card, the ANSYS CDB file makes provision for a wire radius of the segments of type `pipe16` (real constant from the associated RLBLOCK). This is then used during the import and any setting at the IP card is ignored (the IP card radius is still used for filaments of element type 200). For dielectric bodies, one must use an ME card to specify the element type and medium indices. The ANSYS field for the material number cannot be used, since for triangles FEKO requires two such material indices (medium on each side).

The user can also import points from the ANSYS CDB file similar to importing points from FEMAP or NASTRAN files. The points defined in the ANSYS CDB file will then be available in PREFEKO as points (as if they were defined by DP cards) of the form `Cxxx` where `xxx` is the index of the grid point. This may be used, for example, to attach additional structures to the geometry. In addition, the coordinate values of the points are available as variables in PREFEKO. For example, the variables `#c1234x`, `#c1234y` and `#c1234z` are set to the coordinates of the point with index 1234. Note that points are not included by default.

9.2.19 IP card

With this card a number of parameters that define the meshing parameters or also the wire radius are set.


IP - Specify the segmentation parameters
Radius of wire segment
Maximum triangle edge length
Maximum wire segment length
Maximum cuboid edge length
Maximum tetrahedral edge length

Parameters:

Radius of wire segment: Segment radius in m (it is scaled by the SF card if used).

Maximum triangle edge length: Maximum edge length of triangular elements in m (it is scaled by the SF card).

Maximum wire segment length: Maximum segment length for wire segments in m (it is scaled by the SF card).

Maximum cuboid edge length: Maximum edge length of cuboidal volume elements for dielectrics (volume equivalence principle of the MoM) in m (it is scaled by the SF card).

Maximum tetrahdral edge length: Maximum edge length of tetrahedral volume elements (FEM) in m (it is scaled by the SF card).

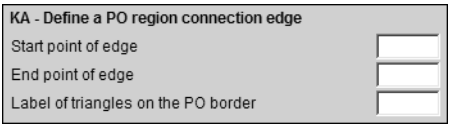
The IP card only affects the commands following it, i.e. it has to be declared prior to the cards that define segments, triangles or cuboids.

It is possible to use more than one IP card in a file. This is necessary when a finer mesh is required in certain parts or when different radii occur in the geometry. For any command (e.g. the BL card) the previous IP card is applicable.

Regarding the meshing, certain rules apply relating the element size to the wavelength etc. see section 2.2.2 for details.

9.2.20 KA card

With this card two points are joined, which form the border of the PO area. On this edge fringe wave currents are taken into account.



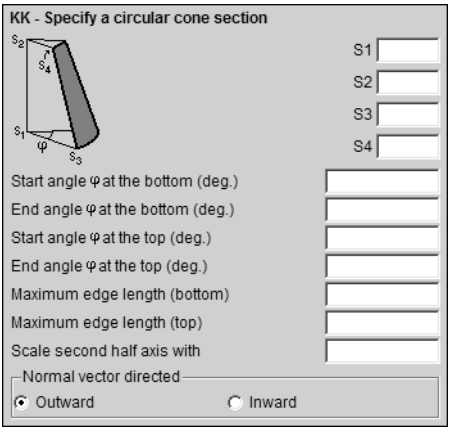
Parameters:

Start point of edge:	Start point of the edge.
End point of edge:	End point. The direction of an edge is arbitrary, i.e. it does not matter which edge point is chosen as the end or start point of the edge.
Label of triangles . . . :	Label of the PO triangles adjacent to the PO border, i.e. the edge correction current from this edge is applied to all triangles with this label.

Note that the surface must be flat, i.e. all triangles with the label specified here must lie in the same plane.

9.2.21 KK card

A mesh of surface triangles in the shape of a conical section can be created with this card. The cone can also be distorted to have an elliptical cross section. Cones or conical sections with included angles that vary from the top to the bottom, or that do not start in a specified plane, can also be created.



Parameters:

S1: The start point of the axis of the cone (the centre of the base).

S2: The end point of the axis (the tip of the cone, or the centre point of the circle when creating a conical section.

S3: A point on the radius of the base.

S4: If this parameter is defined, the cone is cut off at the top; if not the cone has a sharp tip. This point must be in the plane given by S1, S2 and S3.

Start angle at the bottom: The angle φ from the plane S2–S1–S3 at which the bottom of the cone should start.

End angle at the bottom: The angle φ from the plane S2–S1–S3 at which the bottom of the cone should end.

Start angle at the top: The angle φ from the plane S2–S1–S3 at which the top of the cone should start.

End angle at the top: The angle φ from the plane S2–S1–S3 at which the top of the cone should end.

Maximum edge length (bottom): The maximum edge length of the triangles along the base arc — in the plane containing S1 — of the cone. (This value is in m and is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.

Maximum edge length (top): This value only applies if S4 is specified and gives the maximum edge length of the triangles along the top arc — in the plane containing S2 — of the cone. (This value is in m and is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.

Normal vector directed: The triangles can be created such that the normal vector points *Outward* (away from the axis) or *Inward* (to the inside of the cone).

Scale second half axis: If this parameter is empty or is set to 1, a cone with a circular cross section is created. If set to $\frac{b}{a}$, a cone with an elliptical cross section is created. Here $\frac{b}{a}$ gives the ratio of the two half axes, where a is the distance S1–S3. It is recommended to generate elliptical cones with extremely small or extremely large axial ratios with a CAD system as the distortion formulation used in PREFEKO may fail in these cases.

The fineness of the mesh on the shell's surface is determined by the maximum edge length specified by the last IP card prior to the KK card. Along the arcs, accurate modelling of the geometry may require finer segmentation and the values *Maximum edge length (bottom)* and *Maximum edge length (top)* specify the maximum edge length along the corresponding arcs. *Maximum edge length (top)* is only used when a truncated cone is created. If either of these values is not specified the length specified with the IP card will be used on the corresponding arc.

Examples of KK card usage:

All of the following meshes are created using the KK card. These examples show a sharp cone, an oblique elliptical cone, a conical section with different angles at the top and bottom and a conical section where the start angle is not in the plane S2–S1–S3.

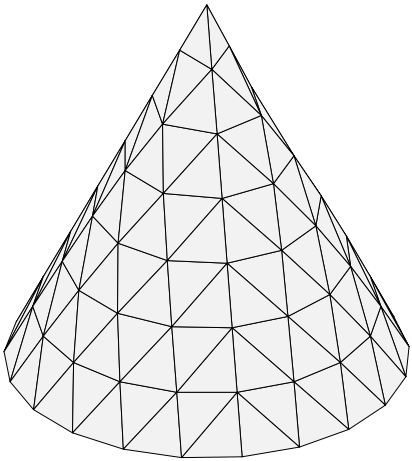


Figure 9-17: Example for the KK card from `demo_KK1.pre`

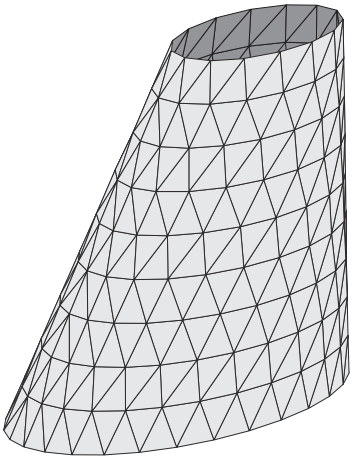


Figure 9-18: Example of a cone with an elliptical cross section from `demo_KK2.pre`

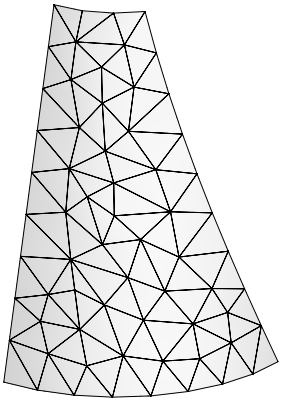


Figure 9-19: Example of a cone with different subtended angles at the top and bottom from `demo_KK3.pre`

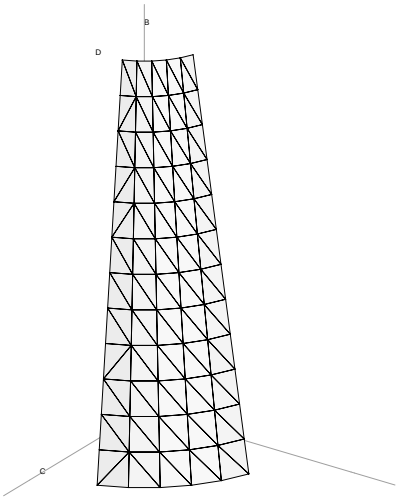
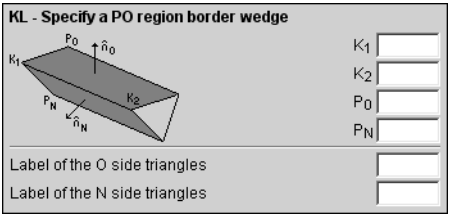


Figure 9-20: Example of a conical section where the start angle is not in the plane defined by S1, S1 and S3 (`demo_KK4.pre`)

9.2.22 KL card

This card defines a wedge for which correction terms are added to the PO currents on the two surfaces connected to it.



Parameters:

K_1: The start point of the edge of the wedge.

K_2: The end point of the edge of the wedge.

P_0: A point on the *o* side of the wedge.

P_N: A point on the *n* side of the wedge.

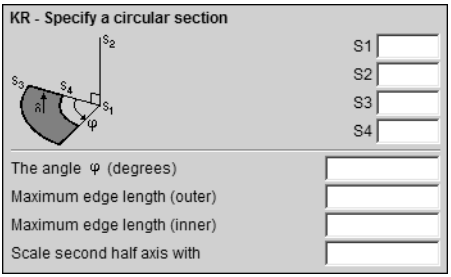
Label of the O side triangles: The label of the PO triangles that are adjacent to the wedge on the *o* side. This means that the corresponding correction term for the *o* side is assigned to the PO triangles that have this label.

Label of the N side triangles: The label of the PO triangles that are adjacent to the wedge on the *n* side. This means that the corresponding correction term for the *n* side is assigned to the PO triangles that have this label.

Note that the wedge must be between flat surfaces, and that all triangles with the label specified here must lie in the same plane.

9.2.23 KR card

This card creates a mesh of surface triangles in the shape of circular region with or without a hole. It is also possible to distort it to an elliptical region.



Parameters:

S1: The centre point of the circle.

S2: A point that is situated at any distance perpendicular to and above the plane of the circle.

S3: A point on the outer arc.

S4: If there is a value present for this parameter, then a circular ring is created. S4 must lie between S1 and S3.

The angle φ: The angle subtended by the arc in degrees.

Maximum edge length (outer): The maximum edge length of the triangles along the outer edge of the arc in m (is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.

Maximum edge length (inner): When a disk with a hole is created, the maximum edge length for the triangles along the inner edge of the arc in m (is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.

Scale second half axis: If this parameter is empty or is set to 1, a circular disk is created. If set to $\frac{b}{a}$, an elliptical disk is created. Here $\frac{b}{a}$ gives the ratio of the two half axes, where a is the distance S1–S3. It is recommended to generate elliptical disks with extremely small or extremely large axial ratios with a CAD system as the distortion formulation used in PREFEKO may fail in these cases.

The circle's plane is perpendicular to the line S1–S2. This length is arbitrary. The radius of the disc is given by the length between the points S3 and S1. The area that is to be subdivided (the shaded region in the figure) is generated by sweeping the edge S3–S1 around the axis S1–S2 through φ degrees in the mathematically positive sense. For $\varphi = 360°$ a circle is obtained.

The fineness of the mesh is determined by the maximum edge length specified by the last IP card prior to the KR card. Along the arcs, accurate modelling of the geometry may require finer segmentation and the maximum edge length values specify the maximum edge length along the outer and inner (if applicable) arcs respectively. If any of these values are not specified the length specified with the IP card will be used on the corresponding arc.

The normal vectors of the triangles on the disk all point in the direction from S1 to S2.

Examples of KR card usage:

The following example meshes are all created using the KR card. Shown is a circular plate, a flat circular ring and a flat elliptical ring.

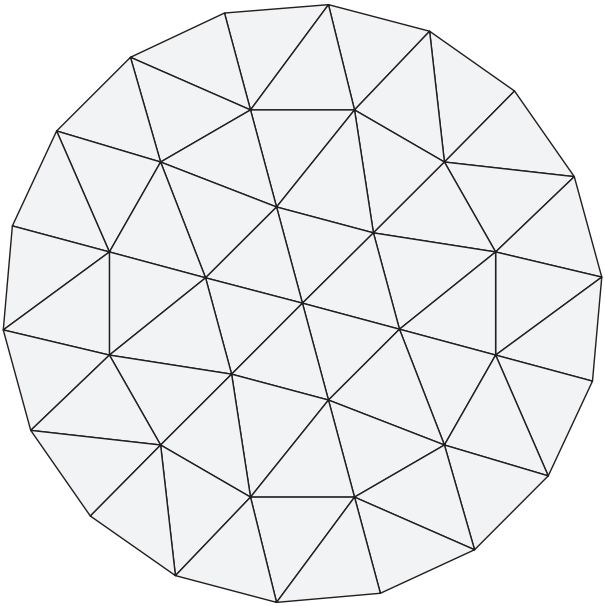


Figure 9-21: First example for the KR card from `demo_KR1.pre`

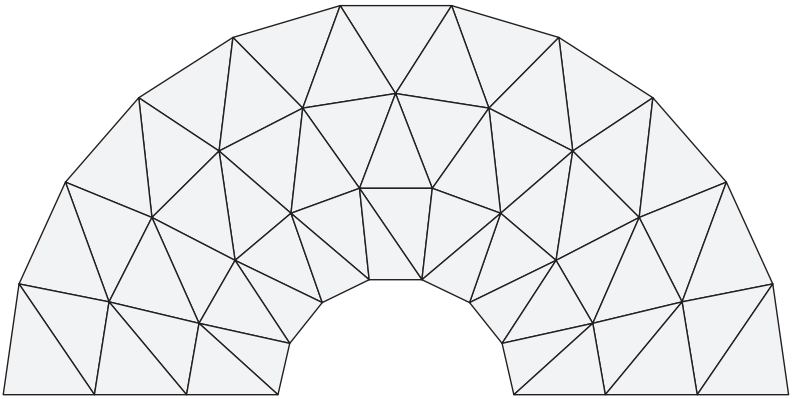


Figure 9-22: Second example for the KR card from `demo_KR2.pre`

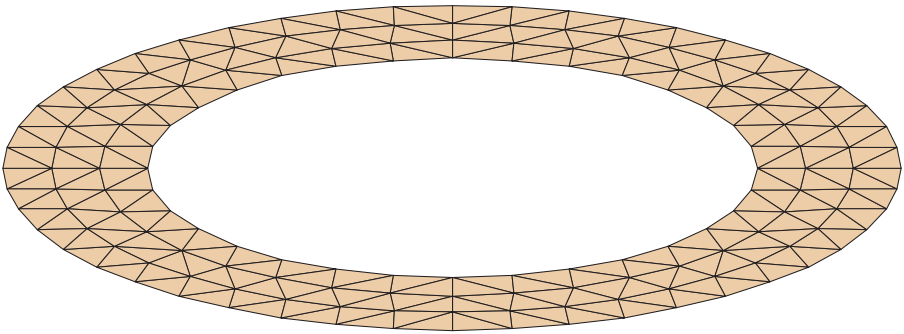
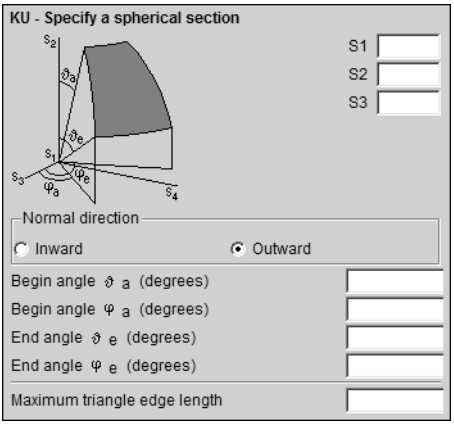


Figure 9-23: Example of an elliptical disk with a hole from `demo_KR3.pre`

9.2.24 KU card

This card creates a mesh of surface triangles in the shape of a spherical section.



Parameters:

S1: The centre of the sphere.

S2: A point that indicates the $\vartheta = 0°$ direction in a spherical coordinate system. The distance between S1 and S2 is the radius of the sphere.

S3: A point that indicates the $\vartheta = 90°$, $\varphi = 0°$ direction in a spherical coordinate system. The distance S1–S3 must be equal to the distance S1–S2.

Normal direction: The triangles can be created such that the normal vectors point *Outward* (away from the centre of the sphere) or *Inward* (towards it).

Begin angle ϑ_a: The start angle ϑ_a in degrees of the spherical segment.

Begin angle φ_a: The start angle φ_a in degrees of the spherical segment.

Begin angle ϑ_e: The end angle ϑ_e in degrees of the spherical segment.

Begin angle φ_e: The end angle φ_e in degrees of the spherical segment.

Maximum triangle edge length: The maximum length of the triangles along the curved edges in m (is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.

A complete sphere may be created with $\vartheta_a = \varphi_a = 0$, $\vartheta_e = 180°$ and $\varphi_e = 360°$.

An example of KU card usage:

The spherical segment shown in figure 9-24 is generated using the KU card.

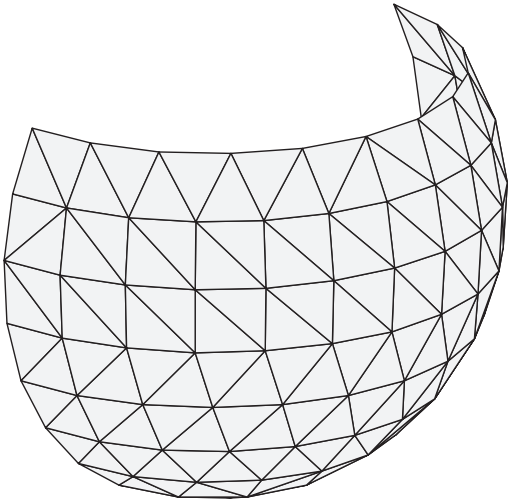


Figure 9-24: Example for the KU card from `demo_KU1.pre`

9.2.25 LA card

With this card, labels are assigned to segments, triangles, polygons, cuboids, UTD cylinders and points.

Parameters:

Label for following . . . : The label assigned to all segments, triangles, etc. defined in cards following this one.

In order to select the position of the feed (Ax cards)[6], the location of impedance loading (LD, LS, LP and LZ cards) or structures on which to apply the skin effect (SK cards), each segment, triangle, etc. is assigned a label. Other applications include the selective transformation or copying of parts of geometry (TG card), and outputting of currents on selected elements (OS card). Labels are also used to define triangles on which to apply physical optics (PO card).

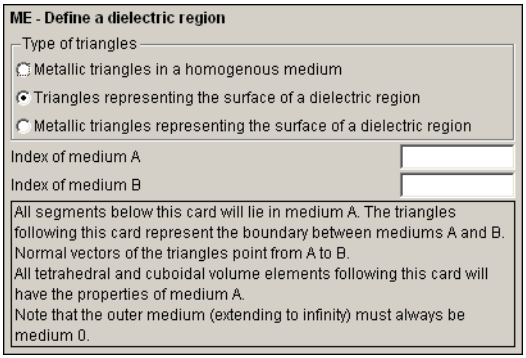
All elements, etc. that are created after the LA card (e.g. by a BP card), are assigned the value specified in the dialog as label. A different label is only assigned by a new LA card. All structures created before the first LA card (or if no LA card is present), is assigned the default label which is 0 (number zero).

Label names can be an arbitrary string using the characters `A-Z`, the digits `0-9` or also the underscore `_`. See section 2.3 for details, also with respect to label increments or label ranges.

[6]The definition Ax stands for any of the control cards A0, A1, A2, ...

9.2.26 ME card

When solving the fields in dielectric objects by means of the MoM/MLFMM surface or volume current methods or by means of the FEM, this card must be used to distinguish the different media or to create segments and metallic triangles within the dielectric. Furthermore, this card is used to switch between the generation of metallic triangles and triangles that represent the surface of the dielectric. Another special case is when metallic triangles represent the surface of a dielectric object (e.g. a dielectric that has been coated with metal).


ME - Define a dielectric region
Type of triangles
Metallic triangles in a homogenous medium
Triangles representing the surface of a dielectric region
Metallic triangles representing the surface of a dielectric region
Index of medium A
Index of medium B
All segments below this card will lie in medium A. The triangles following this card represent the boundary between mediums A and B. Normal vectors of the triangles point from A to B.
All tetrahedral and cuboidal volume elements following this card will have the properties of medium A.
Note that the outer medium (extending to infinity) must always be medium 0.

Parameters:

Metallic triangles in a homogeneous medium: If this option is selected, then all the surface structures between this card and the next ME card are assumed to be fully contained inside the medium specified in the *Medium for triangles/segments* dialog.

Triangles representing the surface of a dielectric body: If this option is selected, then all the surface structures created between this card and the next ME card are assumed to define the boundary between two media. Note that the user needs to provide the names of the medium on both sides of the triangles. The normal vector points from medium A to medium B. Thus if we have a dielectric body of medium `DIELECTRIC`, constructed such that all the triangle normals point outward, we set medium A to for instance `DIELECTRIC` and B to 0 (the number zero always represents the outer free space region).

Metallic triangles representing the surface of a dielectric body: If this option is selected, then all the surface structures created between this card and the next ME card are assumed to define a metallic boundary between two media. The selection of the sides is the same as for the non-metallic case discussed above.

All the wire segments that follow this card are assigned the properties of the medium in which they are found. Triangles are treated differently — it depends upon whether they are metallic triangles or triangles on the boundary of a dielectric object. Here the properties of the media are assigned to the respective sides.

All triangles and segments before a ME card represent metallic structures in free space. This is also the case when an input file does not have a ME card.

When using the FEM and meshing structures into tetrahedral elements or when using the volume equivalence principle in connection with the MoM/MLFMM and meshing into cuboidal volume elements, then the selection for type of triangle is not relevant. The specified medium will be used (medium A if there are multiple media input fields).

The medium name can be an arbitrary string using the characters `A-Z`, the digits `0-9` or also the underscore `_`. See section 2.3 for details, also with respect to ranges etc. (the discussion there for labels applies in a similar manner also to media). Note that the outer medium must always be medium 0 (the number zero).

The use of the ME card to create a simple dielectric sphere is shown in `example_04` (see the *Examples Guide,* note that the normal vectors of the sphere point outwards from medium 1 — the dielectric — to medium 0 — free space). In addition `example_23` shows the more complex example of a dielectric cone (medium 1) mounted on top of a metallic cylinder shown in figure 9-25. There are three types of triangles involved:

- Metallic triangles in free space (Medium 0) on the bottom and side of the cylinder
- Metallic triangles also forming the border surface of the dielectric body on the lid of the cylinder (which is also the basis of the dielectric cone)
- Dielectric triangles forming the surface of the dielectric body (the boundary between medium 1 — the inner dielectric region — and medium 0 — the free space outer region) on the top surface of the cone

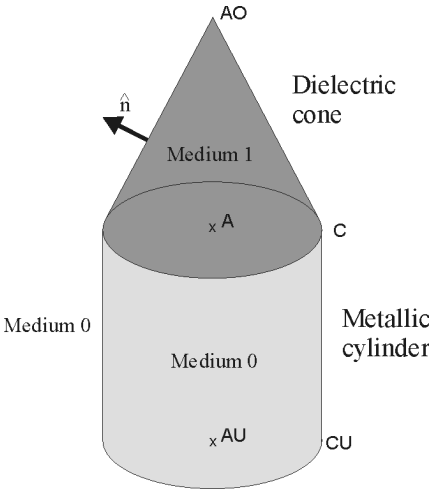
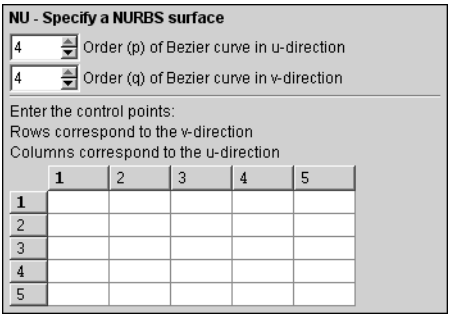


Figure 9-25: Example of a dielectric cone on top of a metallic cylinder, to demonstrate the use of the ME card.

9.2.27 NU card

Surface triangles representing a NURBS surface are created using this card.



Parameters:

Order (p) of Bezier curve in u-direction: The order of the Bézier curve in the $\hat{u}$-direction.

Order (q) of Bezier curve in v-direction: The order of the Bézier curve in the $\hat{v}$-direction.

Both p and q must be in the range from 1 to 4 where 1 is linear, 2 quadratic, and so on. The control points are entered in the table, more or less representing their physical relation. There are p+1 rows and q+1 columns.

It is possible to create a triangular Nurbs surface. In this case all control points on one side must be identical (use the same point). The weights of the control points are specified at the DP card. Note that, for higher order Bézier curves, the surface does not pass through the control points except those on the corners.

Examples of NU card usage:

The "saddle point" shape in figure 9-26 and the linear-quadratic shape in figure 9-27 are generated with NU cards using 4 and 6 control points respectively.

NURBS may also be used to generate flat surfaces with curved edges. The section of a circular plate with a square hole in figure 9-28 is generated using a NU cards.

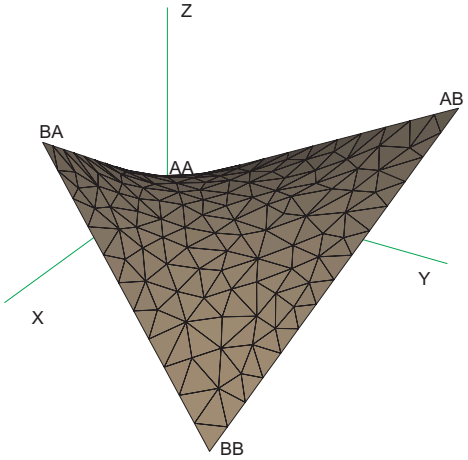


Figure 9-26: "Saddle point" example using the NU card from `demo_NU1.pre`

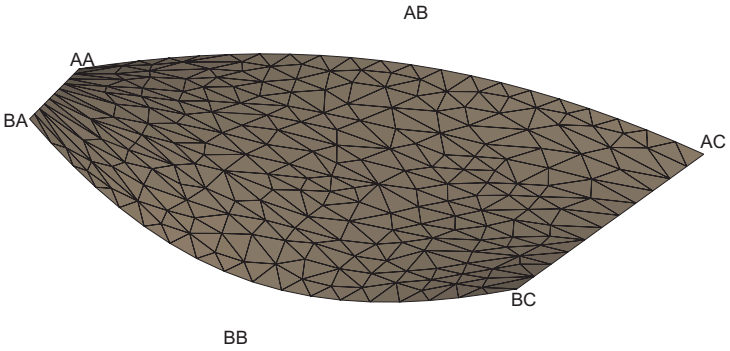


Figure 9-27: Linear-quadratic example using the NU card from `demo_NU2.pre`

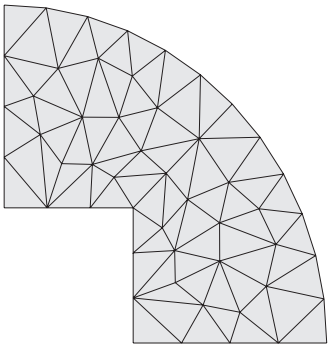
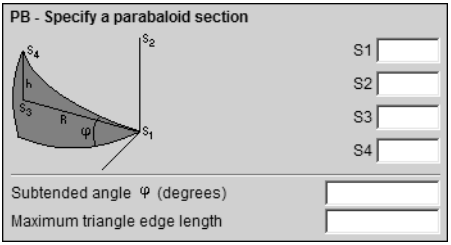


Figure 9-28: A flat surface with curved edges created with an NU card in `demo_NU3.pre`

9.2.28 PB card

This card can be used to generate a section of a parabolic reflector as shown in the figure on the card.



Parameters:

S1: The centre of the paraboloid.

S2: A point perpendicular to the base plane and at any distance above the centre point.

S3: A point on the outer edge of the paraboloid, but on the base plane.

S4: A point in the plane S2–S1–S3, directly above S3 on the edge of the paraboloid.

Subtended angle φ: The angle subtended by the arc of the parabolic reflector in degrees.

Maximum triangle edge length: Maximal edge length of the triangles along the outer edge of the arc in m (is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.

The radius R of the paraboloid is derived from the distance between the points S1 and S3, as can be seen in the figure in the card. The height is determined by the distance between points S3 and S4. The focal point f is determined by:

$$f = \frac{R^2}{4h}.$$

Example of PB card usage:

The parabolic reflector as shown in figure 9-29 can be generated by using the PB card.

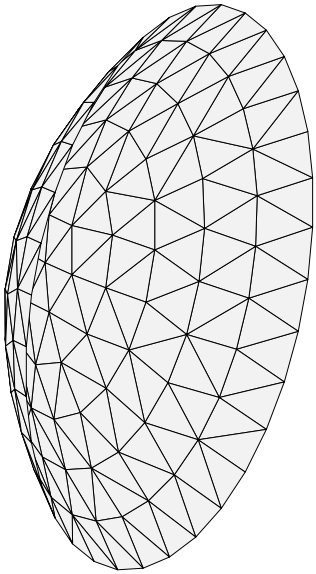
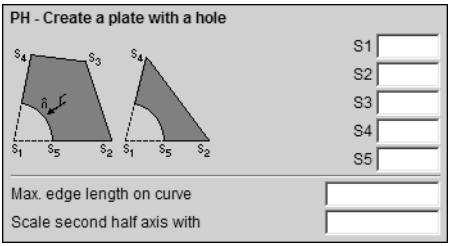


Figure 9-29: Example for the PB card from `demo_PB1.pre`

9.2.29 PH card

This card creates a triangular or quadrangular plate with a circular or elliptical hole as shown in the card.



The hole can be used, for example, to attach a cylinder (ZY card) to the plate and it can be filled with the KR card.

Parameters:

S1: The corner where the hole is located (also the hole's centre).

S2: The second corner of the plate.

S3: The third corner of the plate. If this field is left empty, a triangular plate is created.

S4: The fourth corner of the plate.

S5: A point on the line S1–S2 that forms the starting point of the circle or ellipse bordering the hole. The special case where S5 is identical to S2 is supported, but due to the resulting geometry yields triangles with very small angles

Max. edge length on curve: The maximum edge length of the triangles along the edge of the hole in m (is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.

Scale second half axis: If this parameter is empty or is set to 1, a circular hole is created. If set to $\frac{b}{a}$, an elliptical hole is created. Here $\frac{b}{a}$ gives the ratio of the two half axes, where a is the distance S1–S3. It is recommended to generate the plate with a CAD system if the elliptical hole has an extremely small or extremely large axial ratio as the distortion formulation used in PREFEKO may fail for such cases.

Examples of PH card usage:

The PH card can be used to create the rectangular plate shown in figure 9-30. Note the extremely narrow triangles at the corners as mentioned above. Figures 9-31 and 9-32 show, respectively a quadrangular and a rectangular plate with the same elliptical hole. The triangular plate is obtained by leaving the field S3 empty.

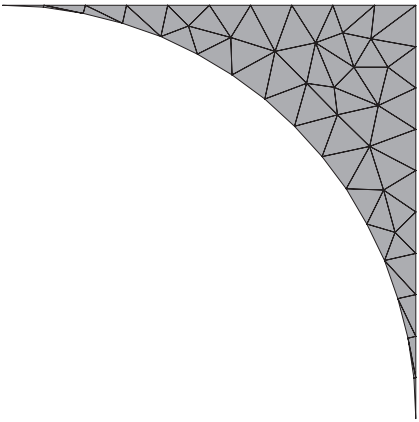


Figure 9-30: Example using the PH card from `demo_PH1.pre`

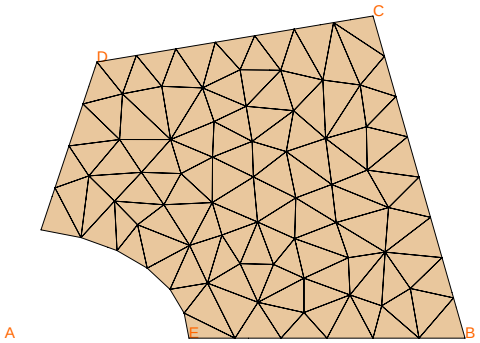


Figure 9-31: Example of a quadrangular plate with an elliptical hole (`demo_PH2.pre`)

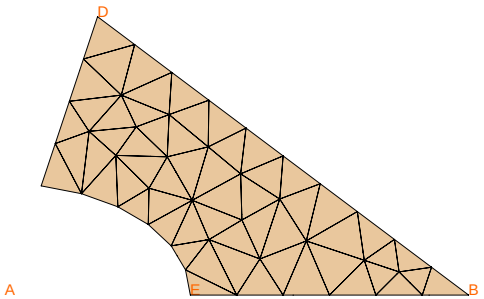
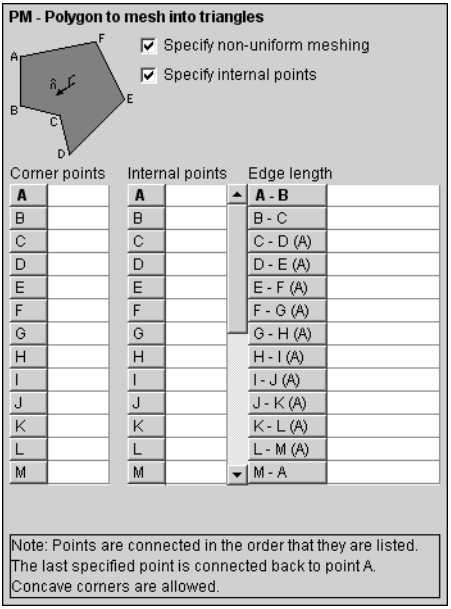


Figure 9-32: Example of a triangular plate with an elliptical hole (`demo_PH3.pre`)

9.2.30 PM card

A surface mesh of triangles in the shape of a polygon is created by using this card. This card also allows the specification of interior mesh points. This card should generally be used in favour of other cards that create flat surface meshes with straight edges.



Parameters:

Specify non-uniform . . . : Check this item to enable the table in which edge lengths for each edge can be entered.

Specify internal points: Check this item to enable the table in which internal points can be specified.

Corner points: Enter the points, previously defined with the DP card, into the rows of this table.

Internal points: Any points internal to the structure where mesh points are required can be entered here. This is particularly useful for the connection points between surfaces and wires.

Edge length: The mesh length on each edge can be set separately in this table. Any blank entries in this table will be meshed with the parameters set in the IP card. There may not be more entries in this table than in the first one.

The polygon is defined by entering the points (previously defined with the DP card) on the corners of the polygon. A maximum of 13 corner points are allowed. The points are connected in the order that they are entered in the PM card. Concave corners are allowed. The user can also specify a smaller or larger mesh size along certain edges.

Examples of PM card usage:

Shown below in figure 9-33 is plate with a concave corner created with the PM card — note the finer mesh along the edges from B to C and C to D.

A plate with three internal points, generated using a PM card is shown in figure 9-34. Note that there are node points at Q1, Q2 and Q3.

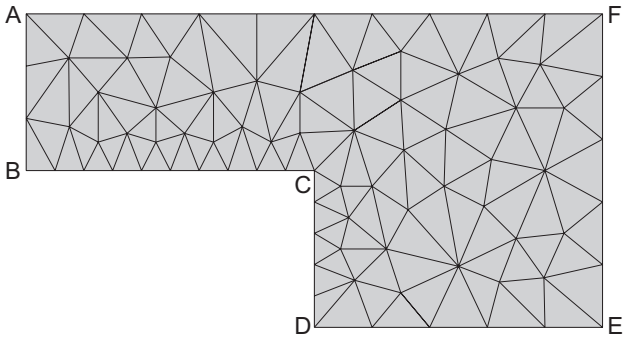


Figure 9-33: Example for the PM card from `demo_PM1.pre`

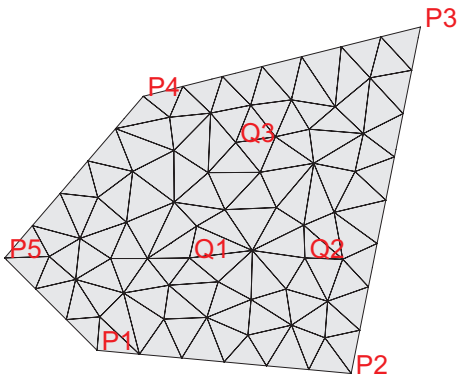
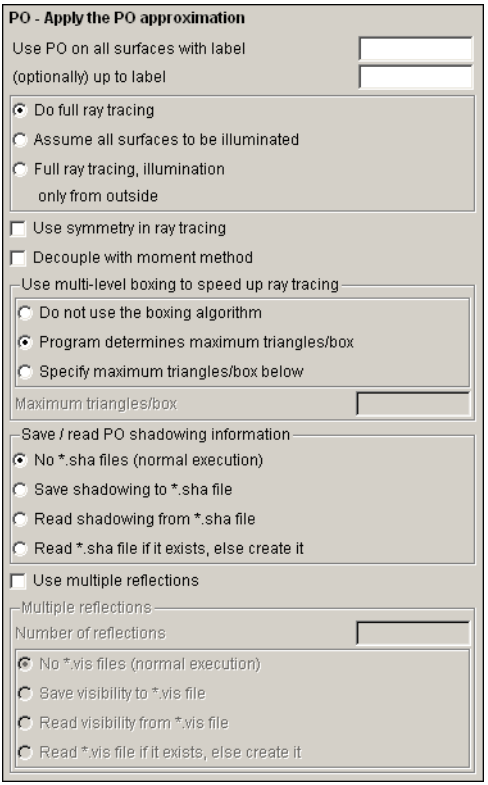


Figure 9-34: Example for the PM card with internal mesh points from `demo_PM2.pre`

9.2.31 PO card

With this card the application of the physical optics approximation is possible.


PO - Apply the PO approximation
Use PO on all surfaces with label
(optionally) up to label
Do full ray tracing
Assume all surfaces to be illuminated
Full ray tracing, illumination only from outside
Use symmetry in ray tracing
Decouple with moment method
Use multi-level boxing to speed up ray tracing
Do not use the boxing algorithm
Program determines maximum triangles/box
Specify maximum triangles/box below
Maximum triangles/box
Save / read PO shadowing information
No *.sha files (normal execution)
Save shadowing to *.sha file
Read shadowing from *.sha file
Read *.sha file if it exists, else create it
Use multiple reflections
Multiple reflections
Number of reflections
No *.vis files (normal execution)
Save visibility to *.vis file
Read visibility from *.vis file
Read *.vis file if it exists, else create it

Parameters:

Use PO on all surfaces with label: together with *(optionally) up to label* are used to specify the label or range of labels of all metallic / dielectric triangles and polygons that are treated with the physical optics approximation. If the second field is left blank, only the label specified in the first field is considered. See also LA and CB cards and also general discussion of label ranges in section 2.3.

Do full ray tracing: Normal, complete ray tracing is carried out.

Assume all surfaces to be illuminated: The ray tracing is switched off to save computational time. The assumption is made that all triangles on which the PO approximation is made are illuminated by the source and the moment method area. The side in relation to the normal vector is automatically determined.

Full ray tracing, illumination only from outside: Full ray tracing is done, but metallic triangles can only be lit from the side to which the normal vector is pointing. See note below.

Use symmetry in ray tracing: If full ray tracing is done, then symmetry can be used to reduce the computational time required to determine the shading. For electric and magnetic symmetry, this speedup is always used. If geometrical symmetry is used, then this item should be checked to utilise symmetry. It is possible to e.g. define half a plate and create the other half through geometric symmetry. An asymmetric object may then be placed in front of the plate. In this case symmetry should not be used in the ray tracing.

Decouple with moment method: When this item is unchecked, the coupling between the MoM region and the PO region is neglected. The implication is that the currents in the PO region has no effect on the current distribution in the MoM region. This option, which should lead to some saving in computational time and storage space, is especially useful when the PO region and the MoM is not directly adjacent.

Use multi-level boxing to speed up ray tracing: The ray tracing required for the physical optics is accelerated by recursively subdividing the problem domain, a so called "multilevel tree". It must balance memory requirement against speedup — both increase as the number of levels increases. The number of levels is determined by specifying the number of triangles at the lowest level. The user can specify not to use this algorithm, for the program to determine maximum triangles/box or to specify this number manually. When a number is specified manually, it should be greater than 2 and at least a factor 10 less than the number of triangles in the problem. In general these options should only be set by advanced users.

Save/read PO shadowing information: For the PO formulation the information which triangles are illuminated and which are shadowed from the sources is required. Normally FEKO computes this each time, but there is an option to store this to a file and load again to save time when the geometry stays constant (say for multiple runs using different frequencies). The options are:

- *No *.sha files (normal execution):* Shadowing information is not stored or read – the default behaviour.
- *Save shadowing to a *.sha file:* The PO shadowing information is written to a `*.sha` file for later reuse.
- *Read shadowing from a *.sha file:* The PO shadowing information is read from the `*.sha` file, i.e. the ray-tracing part is skipped. For large models this can result in considerable time saving.

- *Read *.sha file if it exists, else create it:* If a `*.sha` file exists, the PO shadowing information is read from this file. Otherwise the information is calculated and saved in a `*.sha` file for later use.

Use multiple reflections: When this item is checked, multiple reflections are considered for the ray tracing. The number of reflections that must be considered is set in the *Number of reflections* dialog. This parameter determines the number of reflections to be taken into account for triangles with labels in the specified range. (For example, the *Number of reflections* must be at least 2 to calculate the scattering from a dihedral and at least 3 for a trihedral.) Increasing the number of reflections that must be considered significantly increases computation time, and this should only be done based on physical considerations.

Visibility information: The visibility information related to multiple reflections can be saved to reduce the computation time for future runs. There are four options that can be selected with respect to saving the multiple reflection visibility information:

- *No *.vis files (normal execution):* Visibility information is not used or stored – the default behaviour.
- *Save visibility to a *.vis file:* The PO visibility information is stored in a `*.vis` file for later reuse.
- *Read visibility from a *.vis file:* The PO visibility information is read from the `*.vis` file, i.e. the calculation of the visibility information is skipped. For large models this can result in considerable time saving.
- *Read *.vis file if it exists, else create it:* If a `*.vis` file exists, the PO visibility information is read from this file. Otherwise the information is calculated and saved in a `*.vis` file for later use.

The physical optics (PO) approximation can only be used for certain structures. Structures where the antenna is situated in front of a reflector are well suited. Then PO can be used for the triangles that form the reflector. This results in a large reduction in computational time and memory for electrically large objects.

Note that the ray tracing options and the number of reflections can be specified on a per label basis, by using multiple PO cards. All other parameters can only be specified once. For the global parameters, the values of the last PO card will be used.

Using *Full ray tracing, illumination only from outside* has two main applications:

- Acceleration of the PO ray tracing with closed bodies (the normal vector must then point outward), since the dot product of the normal and propagation vectors can be

used to quickly determine if a triangle or polygon is to be used in the ray tracing. In this case the closed model must be constructed with the normals pointing outward.

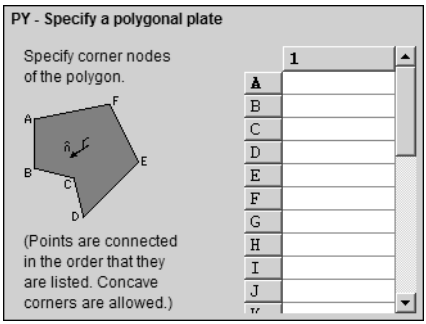
- In, for example the MoM/PO hybrid method on a closed body, the MoM region (such as an antenna) can be prevented from illuminating the PO region from inside.

A basis function that has been assigned to an edge between two triangles will only be solved with the PO, if the PO approximation has been declared for the labels of both triangles.

The metallic PO region must be perfectly conducting, i.e. no losses are allowed. Dielectric coatings (see CO card) and thin dielectric sheets (see SK card) can, however, be treated with the PO approximation.

9.2.32 PY card

This card defines (by specifying the corner points) a polygonal surface to which the UTD formulation is applied.



Parameters:

A:	The first corner point of the polygon.
B:	The second corner point of the polygon, etc.

A maximum of 26 corner points are allowed. The points are connected in the order that they are entered in the PY card. The corner points have to be defined prior to the PY card by a DP card.

Example of PY card usage:

This card can be used to generate the polygon (in this case a triangle) shown in figure 9-35. Note that this triangle is not meshed, as the result would be if the BT or PM cards had been used.

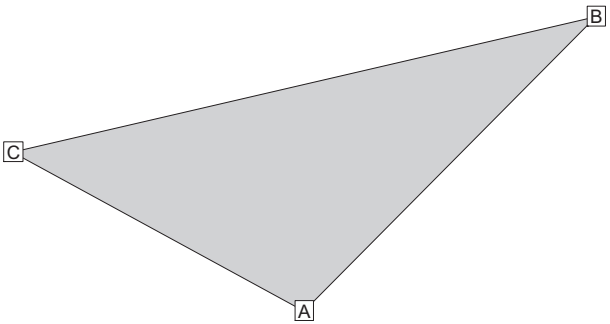
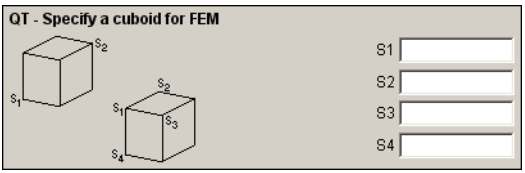


Figure 9-35: Example for the PY card from `demo_PY1.pre`

9.2.33 QT card

This card is used to create a dielectric or magnetic cuboid, meshed into smaller tetrahedral volume elements, for solutions using the FEM. The meshing parameters as set at the IP card are used, and the medium as set at the ME card is assigned to all created tetrahedral elements.



Parameters:

S1:	First corner of the cuboid.
S2:	Opposite corner of the cuboid if aligned with the principle planes, otherwise one of the corners adjacent to the first corner.
S3:	Optional third corner of the cuboid, adjacent to the first.
S4:	Optional fourth corner of the cuboid, adjacent to the first.

Note that when using the FEM, in the same model metallic structures are allowed (metallic surfaces also inside the FEM region or on the FEM boundary, wires only outside). But using dielectric bodies inside the MoM region at the same time is not supported.

Example of QT card usage:

The dielectric cuboid shown in figure 9-36 is generated using a QT card.

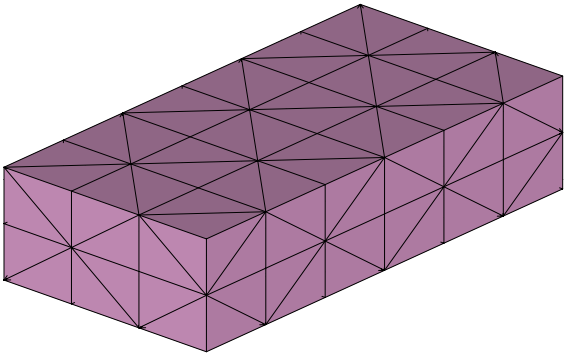
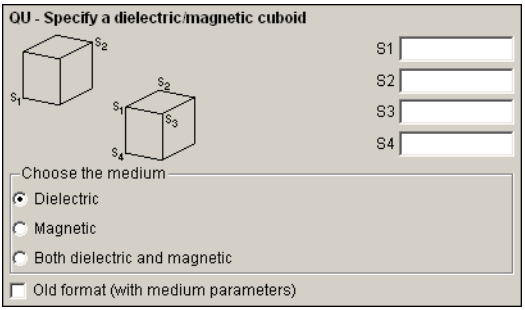


Figure 9-36: Example for the QT card from `demo_QT1.pre`

9.2.34 QU card

This card is used to create a dielectric or magnetic cuboid, meshed into smaller cuboidal volume elements, for solutions using the volume equivalence principle in the MoM. The meshing parameters as set at the IP card are used, and the medium as set at the ME card is assigned to all created cuboidal elements.



Parameters:

S1: First corner of the cuboid.

S2: Opposite corner of the cuboid if aligned with the principle planes, otherwise one of the corners adjacent to the first corner.

S3: Optional third corner of the cuboid, adjacent to the first.

S4: Optional fourth corner of the cuboid, adjacent to the first.

Choose the medium: Select here whether the cuboid is dielectric or magnetic or both (this is always with respect to the environment, e.g. if the relative permittivity ε_r of the cuboid material differs from the environment, then this is a dielectric cuboid).

Old format (with medium parameters): Up to and including FEKO Suite 4.3 for cuboids the material parameters were specified directly at the QU card. This is the old card format. From FEKO Suite 5.0 onwards the concept of ME/DI cards is used to define the material by name and to set the material parameters. When checking this option, the panel layout will change to the old format so that the material parameters can be entered (depending then on the selection whether dielectric or magnetic cuboid). FEKO then uses a compatibility mode and creates artificial media with names `QU_MED_xx` (`xx` is an index). It is not recommended to use the old card format for new models. When working on old models and pressing F1 in EDITFEKO on an existing QU card, the old format panel will be opened automatically.

Dielectric bodies treated with the volume equivalence principle (using cuboids) cannot be used simultaneously with dielectric bodies treated with the surface equivalence principle or the FEM or with special Green's functions.

Example of QU card usage:

The dielectric cuboid, shown in figure 9-37 is generated using a QU card.

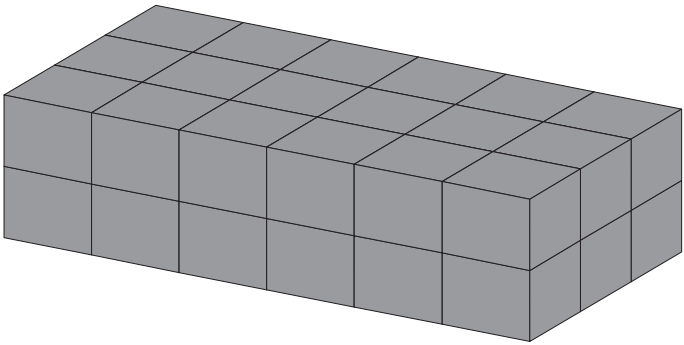


Figure 9-37: Example for the QU card from `demo_QU1.pre`

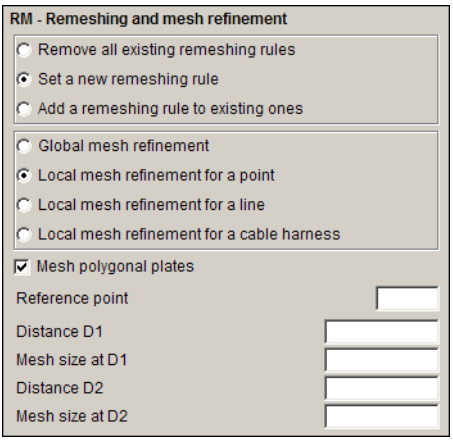
9.2.35 RM card

The RM card provides a sophisticated remeshing and adaptive mesh refinement facility. Most types of meshes (surface mesh with triangular patches, wire segment mesh, cuboidal volume elements) created by any option supported in FEKO (e.g. direct creation in PREFEKO with cards, but also import from NASTRAN, FEMAP, PATRAN etc.) can be used as a basis, and one can then apply either a local or a global mesh refinement. Unfortunately in FEKO Suite 5.0 there is still a restriction that tetrahedral volume elements as used for the FEM cannot be refined with the RM card.

A local mesh refinement refers to a point or a line as reference, or also to a complex cable harness geometry, which is defined directly by importing the corresponding `*.rsd` file from CableMod.

Note that similar to other FEKO cards, the RM card applies only to what follows in the `*.pre` file after the line where the RM card has been read. So for instance if one wants to import a mesh from a NASTRAN file via the IN card and do a mesh refinement during the import, then one first has to use the RM card, then followed by the IN card.

Multiple RM cards can be used, for instance if there are multiple areas in a model where the mesh shall be refined locally. Or also if we use a mesh refinement with respect to one point, the mesh size increases linearly with distance, and by adding another RM card with a global mesh refinement option, a threshold can be set.


RM - Remeshing and mesh refinement
Remove all existing remeshing rules
Set a new remeshing rule
Add a remeshing rule to existing ones
Global mesh refinement
Local mesh refinement for a point
Local mesh refinement for a line
Local mesh refinement for a cable harness
Mesh polygonal plates
Reference point
Distance D1
Mesh size at D1
Distance D2
Mesh size at D2

Parameters:

Remove all existing . . . : Clear all previously defined remeshing rules (i.e. the behaviour is as if no RM card was read). This option is useful if after having imported a structure using mesh refinement, one wants to import another structure or create objects directly in PREFEKO, and for these new structures no mesh refinement shall be used. If this option is checked, all the other parameters are ignored.

Start a new remeshing rule: Set a new remeshing option (previously read RM cards will be discarded).

Add a remeshing rule: Add a remeshing rule to the already defined ones (i.e. existing RM card rules will be kept, the new rule will be added to these).

Global mesh refinement: Global mesh refinement using the specified finer mesh size.

Local mesh refinement for a point: Here an adaptive mesh refinement is performed to obtain a finer mesh close to a point. The point must have been defined before with a DP card, and its name is passed in the input field for the reference point. Note that this point can be located arbitrarily in space, there is no need for this to be on the meshed structure.

Local mesh refinement for a line: Here an adaptive mesh refinement is performed to obtain a finer mesh close to a line. The line is defined by its start and end point. These two points must have been created before with DP cards. Multiple simultanesously active RM cards can be used to perform a mesh refinement with respect to an arbitrary polygonal shaped line, composed by multiple straight line segments.

Local mesh refinement for a cable harness: With this option one can perform a local mesh refinement close to a cable harness. The cable harness geometry is specified by a CableMod `*.rsd` file. The filename of this must be entered into the respective input field (visible only when this option has been selected).

Mesh polygonal plates: As a special feature, the RM card also allows to mesh unmeshed polygonal plates (which are used in FEKO for the UTD) during the import. This can be very useful if e.g. a UTD model is imported from FEMAP using then boundary surfaces, and instead of the UTD a MoM or MLFMM or PO solution shall be conducted (where a mesh is required).

Reference point: When using local mesh refinement with respect to a point, then here the name of this point is entered (the point must have been specified before at a DP card).

Start point of line: When using local mesh refinement with respect to a line, then here the name of the start point of the line is entered (the point must have been specified before at a DP card).

End point of line: When using local mesh refinement with respect to a line, then here the name of the end point of the line is entered (the point must have been specified before at a DP card).

*CableMod *.rsd file:* When using local mesh refinement with respect to a cable harness, then here the filename of the CableMod `*.rsd` file is specified.

Global finer mesh size:	When a global mesh refinement is used, then this is the new mesh size which shall be applied. Mesh coarsening is not supported, only mesh refinement. So when the new mesh size is larger than the existing mesh size, simply no mesh refinement will be done.
Distance D1:	Reference distance d_1 for the mesh refinement, discussed below.
Mesh size at D1:	Mesh size s_1 at the reference distance d_1, discussed below.
Distance D2:	Reference distance d_2 for the mesh refinement, discussed below.
Mesh size at D2:	Mesh size s_2 at the reference distance d_2, discussed below.

The mesh sizes specified for the global or local mesh refinement apply to all types of geometry (i.e. triangles, wires, cuboidal volume elements etc.) in the same manner. This is not a principal restriction. If different refinement options are desired say for wires and triangles, one can use one RM card, create or import say just triangles, and then use another RM card and after this create or import just wires etc.

When the actual remeshing is done, then for each created or imported mesh element (e.g. a triangular surface patch element) PREFEKO internally loops over all the RM cards which are active at this time, and determines the local mesh length for each RM card, and the smallest of these is then used for the actual mesh refinement of the specific mesh element under consideration.

If one RM card specifies a global mesh refinement, then the local mesh size is readily given by the global finer mesh size. If one does local mesh refinement with respect to a point, then first the distance of the mesh element to this point is determined. Similarly for a line or a cable harness, the shortest distance from the mesh element to this line or cable is determined. If we assume that this distance is d, then the local mesh size s is given by the equation

$$s = s_1 + \frac{s_2 - s_1}{d_2 - d_1}\,(d - d_1) \quad .$$

This means that for a distance $d = d_1$ we get the mesh size $s = s_1$, and for the distance $d = d_2$ the mesh size is $s = s_2$. For any other distances (smaller than d_1, in between d_1 and d_2, or also larger than d_2) a linear interpolation is used by means of the formula above. Thus the linear increase of the mesh size also continues for larger distances, but one should keep in mind that the RM card can only do a mesh refinement and no mesh coarsening, i.e. as soon as for larger distances the remeshing option exceeds the already used mesh size of the original model, simply nothing will happen. Although not required, it is often useful to set the mesh size s_2 identical to the global already existing mesh size, then the parameter d_2 readily controls the region where a local mesh refinement is desired (i.e. for distances d larger than d_2 the original mesh will be kept).

It shall also be mentioned here that if a CableMod `*.rsd` file is imported, in order to evaluate distance d between each mesh element and the cable harness in the right base

unit (if an SF card scaling factor is set this can be for instance mm), the cable harness coordinates have to be scaled accordingly. Thus the SF scaling factor must be known before the RM card can be used. PREFEKO will give an error if an SF card is read and a RM card was processed before. The user must then just move the SF card in front of the RM card in the `*.pre` file.

Examples of RM card usage:

A first example is shown in figure 9-38 with the original mesh on the left hand side and on the right hand side the result of a local mesh refinement with respect to a point is given. For the example in figure 9-39 a local mesh refinement with respect to two lines is used (i.e. two simultaneously active RM cards).

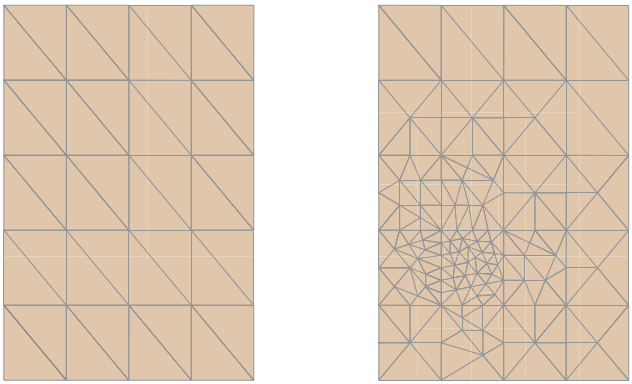


Figure 9-38: Example of local mesh refinement with respect to a point from `demo_RM3.pre`

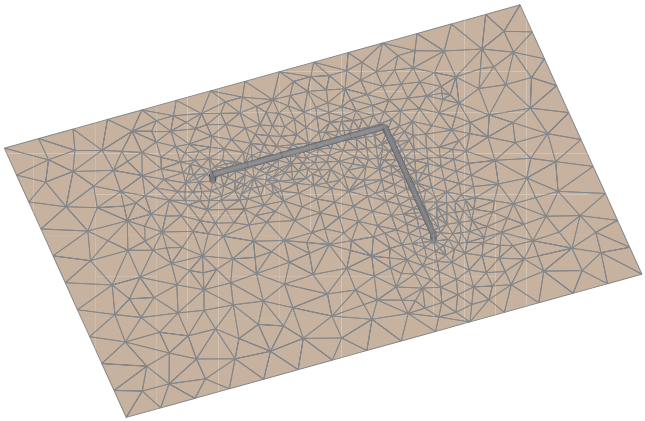
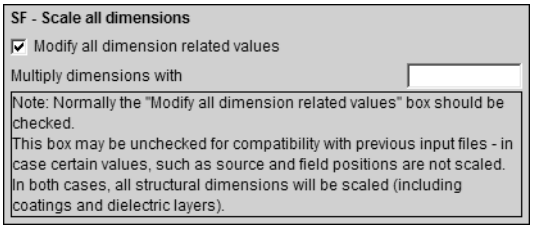


Figure 9-39: Example of local mesh refinement with respect to lines `demo_RM5.pre`

9.2.36 SF card

With this card the scaling of the geometric data is possible. This is useful for specifying models in a convenient unit (e.g. cm) and specifying a scaling factor once, since internally FEKO uses all dimension related values in metres.


SF - Scale all dimensions
Modify all dimension related values
Multiply dimensions with
Note: Normally the "Modify all dimension related values" box should be checked.
This box may be unchecked for compatibility with previous input files - in case certain values, such as source and field positions are not scaled.
In both cases, all structural dimensions will be scaled (including coatings and dielectric layers).

Parameters:

Modify all dimension related values: If this item is checked all geometrical dimensions are scaled. If unchecked some coordinate values are not scaled (for example the positions of near field calculations, see the list below). This should only be unchecked for backwards compatibility with old input files.

Multiply dimensions with: The scale factor. For example, if this is set to 0.001, all dimensions are entered in mm.

Only one SF card is allowed in the input file. This is global and can be positioned anywhere. (However, since it is a geometry card it must be before the EG card.)

If *Modify all dimension related values* is unchecked, the following is scaled:

- Coordinates of the corner points of the triangular surface elements
- Coordinates of the corner points of the segments
- Radii of the segments
- Coordinates of the corner points of the cuboids
- Radii of the all the layers when the Green's function for a homogeneous or layered dielectric sphere is used (GF card, section 10.2.30)
- Thickness of the layers when the Green's function for a planar, multilayered substrate is used (GF card, section 10.2.30)
- Coordinates of the corner points of the polygonal plates
- Coordinates, radii and dimensions of UTD cylinders

- Coordinates of the corner points of tetrahedral volume elements
- Thickness of dielectric surface elements (SK card, section 10.2.42)
- Radius and thickness of a wire coating (CO card, section 10.2.23)
- Coordinates of wedges and edges in the PO region
- Coordinates of the Fock region
- Transmission line length and end point coordinates (TL card, section 10.2.44)
- The dimensions of the aperture used in the AP card and the amplitudes of the A5 and A6 dipoles which depend on the incremental areas.
- The variable *Maximum identical distance,* specified with the EG card (section 9.2.12), which controls whether two points are considered to lie at the same point in space

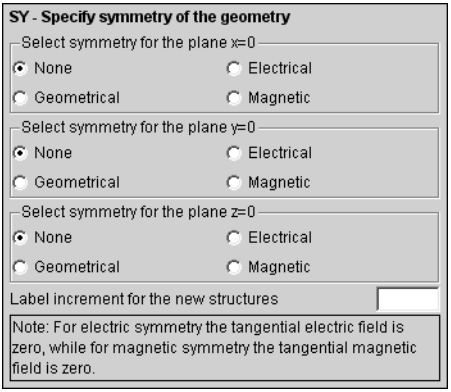
If it is checked all geometrical dimensions and coordinates are scaled. This includes, in addition to the parameters listed above, the following:

- The coordinates of the source point specified in the excitation cards A1, A2, A3, A4, A5, A6, A7 (if the selection is not made by label).
- Coordinates of the origin of the radiation pattern specified with the AR card.
- Coordinates of the start and end points of the impressed currents for the AI and AV source cards, as well as the wire radius specified with these cards.
- Radii of the coaxial feed in the A3 card.
- Radius of the approximated connecting segment in the A4 and L4 cards.
- Positions where the near field is calculated with the FE card.
- Offset in the near field calculation (OF card, section 10.2.37).

Note that if, for example, all data is specified in mm with the scaling set to 0.001, all input values are interpreted as mm. This also applies to the segmentation parameters (IP card) and possible translations (TG card).

9.2.37 SY card

Here symmetry can be used to generate the geometry and to reduce computation time.



Parameters:

Select symmetry for the plane $x = 0$: The type of symmetry, if any, in the yz-plane.

Select symmetry for the plane $y = 0$: The type of symmetry, if any, in the xz-plane.

Select symmetry for the plane $z = 0$: The type of symmetry, if any, in the xy-plane.

Label increment for the new structures: After they are mirrored, the labels of the new elements are incremented with the value specified in this field. Label 0 is, however, not incremented. The corresponding new elements will also have label 0. If this field is empty or set to 0, the labels are not incremented, i.e. the new elements will have the same label as the one they were created from.

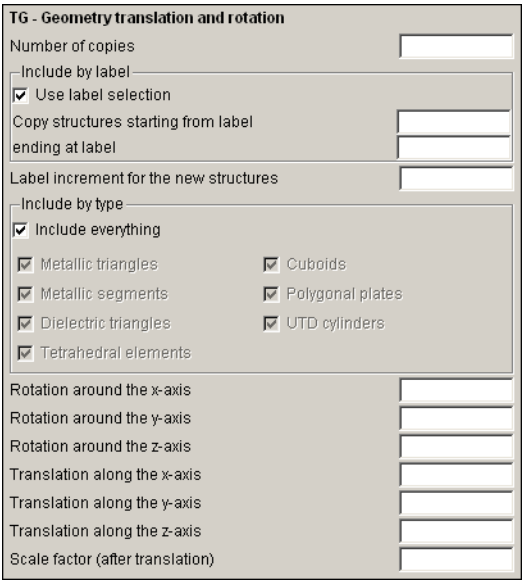
All the conducting and/or dielectric triangles, segments, cuboids, tetrahedral volume elements, wedges, edges, Fock regions and polygonal surfaces that have been declared before the SY card, are mirrored. Furthermore, the second and third corners of the triangles are swapped, such that the direction of the normal vector is retained. Likewise the corners of image polygons are rearranged to retain the normal direction. (The first corner point of the original polygon becomes the last corner of the mirror image.)

Sources are not mirrored. If, for example, a Hertzian dipole is placed on one side of the symmetry plane, the user must also place the correct image on the opposite side of the symmetry plane.

Multiple SY cards can be used and it is possible to mirror around more than one plane at once. A detailed description of the different types of symmetry (geometric, electric and magnetic symmetry) was given in section 2.4.

9.2.38 TG card

With this command the already entered geometric elements (triangles, segments, etc.) can be translated, rotated and/or scaled. It is also possible to duplicate structures.



Parameters:

Number of copies: The number of copies to make, for example if set to 3 the selected elements will be rotated / translated / scaled 3 times such that there will be a total of 4 structures. If set to 0, the existing elements, are translated / rotated and the number of elements remains the same.

Use label selection: If this option is not checked, then the TG card applies to all the previously defined geometry. If this option is checked, then a label selective processing is possible.

Copy structures starting from: together with *ending at label* can be used to apply the TG card only to a selected part of the structure. The TG card is applied only to those elements whose label lies within the range set here (see also LA and CB cards and also general discussion of label ranges in section 2.3). If the second field is left empty, only structures with the label set in the first field are considered. Note that certain element types on the specified label(s) can be excluded from the selection lower in the card.

Label increment for the new structures: Each newly generated structure will be assigned a label that is incremented by this value from that of the original structure. An exception is the label 0 which is retained.

Include: This group can be used to specify which element types (provided they satisfy the label criterion) are rotated / translated.

Rotation around the x-axis: Angle of rotation α_x around the x axis in degrees.

Rotation around the y-axis: Angle of rotation α_y around the y axis in degrees.

Rotation around the z-axis: Angle of rotation α_z around the z axis in degrees.

Translation along the x-axis: Translation Δ_x in the x direction in m (scaled by SF card).

Translation along the x-axis: Translation Δ_y in the y direction in m (scaled by SF card).

Translation along the x-axis: Translation Δ_z in the z direction in m (scaled by SF card).

Scale factor: The scaling factor γ, with which the structures must be scaled. (If left empty, it defaults to 1.) For wire segments the wire radius is scaled as well as the coordinates of the start and end points.

When an SY card (symmetry) is used before the TG card, the TG card resets the symmetry if the new structures invalidates the symmetry. Cases where the symmetry is not reset is when, for example, the plane $z = 0$ is a symmetry plane and the TG card specifies rotation about the z axis for a symmetrical selection of elements. In this case the symmetry is retained.

If more than one copy is made, successive points are generated from the previous point using the same relation.

With a TG card the simultaneous rotation around multiple axes as well as translation in multiple directions is possible. A point $(x,\ y,\ z)$, for example the corner point of a triangle, is transformed to a new point

$$\begin{bmatrix} x_T \\ y_T \\ z_T \end{bmatrix} = \gamma\, M \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \gamma \begin{bmatrix} \Delta_x \\ \Delta_y \\ \Delta_z \end{bmatrix}$$

with the rotation matrix

$$M = \begin{bmatrix} \cos\alpha_y \cos\alpha_z & -\cos\alpha_y \sin\alpha_z & \sin\alpha_y \\ \cos\alpha_x \sin\alpha_z + \sin\alpha_x \sin\alpha_y \cos\alpha_z & \cos\alpha_x \cos\alpha_z - \sin\alpha_x \sin\alpha_y \sin\alpha_z & -\sin\alpha_x \cos\alpha_y \\ \sin\alpha_x \sin\alpha_z - \cos\alpha_x \sin\alpha_y \cos\alpha_z & \sin\alpha_x \cos\alpha_z + \cos\alpha_x \sin\alpha_y \sin\alpha_z & \cos\alpha_x \cos\alpha_y \end{bmatrix}$$

Multiplication by the rotation matrix M effectively rotates a point first by an angle α_z around the z axis, then by an angle α_y around the y axis and finally by an angle α_x around x axis. It is important to note that the second rotation around the y axis represents the

currently valid y axis and not the original one (i.e. the one from the first rotation around the z axis). Similar for the third rotation around the x axis.

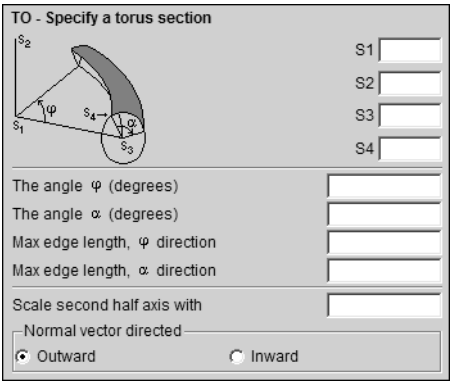
The transformation angles as used by FEKO in this order are generally referred to as Kardan angles as opposed to the also commonly used Euler angles. If the rotation shall be performed in the other order (i.e. first around the x axis, then around the y axis and finally around the z axis), then one can simply use multiple consecutive TG cards. But since the same rotation algorithm is also used at other FEKO cards (for instance AC or AR) where one cannot use multiple cards, a short PREFEKO code segment shall be given here which illustrates how the angles can be converted:

```
** Desired rotation angles such that we rotate first around x, then y, and
** then around z
#a1 = 30   ** Angle in deg. around x axis
#b1 = 60   ** Angle in deg. around y axis
#c1 = 90   ** Angle in deg. around z axis

** Precompute some sin() and cos() terms
#ca1 = cos(rad(#a1))
#cb1 = cos(rad(#b1))
#cc1 = cos(rad(#c1))
#sa1 = sin(rad(#a1))
#sb1 = sin(rad(#b1))
#sc1 = sin(rad(#c1))

** Some auxiliary terms resulting from equating the transformation matrices
#cc2 = #cb1*#cc1/(sqrt((#cb1*#cc1)^2+(#ca1*#sc1-#sa1*#sb1*#cc1)^2))
#cb2 = #cb1*#cc1/#cc2
#ca2 = #ca1*#cc2/#cc1
#sa2 = #cc2*(#sa1*#cc1-#ca1*#sb1*#sc1)/(#cb1*#cc1)
#sb2 = #sa1*#sc1+#ca1*#sb1*#cc1
#sc2 = #cc2*(#ca1*#sc1-#sa1*#sb1*#cc1)/(#cb1*#cc1)

** Finally compute the angles which must be used in FEKO in the TG card
** for the rotation order first around z, then around y, and then around x
#a2 = deg(atan2(#sa2,#ca2))
#b2 = deg(atan2(#sb2,#cb2))
#c2 = deg(atan2(#sc2,#cc2))
```

The file card based version of `example_18.pre` (see the *Examples Guide*) gives an example of an application of the TG card.

9.2.39 TO card

Using this card a surface mesh in the form of a toroidal segment can be generated.



Parameters:

S1: The centre of the toroid.

S2: A point that is perpendicular and is situated an arbitrary distance above the plane of the toroid.

S3: The start point of the axis of the toroid.

S4: A point on the surface of the toroidal segment. It must be in the plane S2–S1–S3.

The angle φ: The angle of rotation around the axis S1–S2.

The angle α: The angle of rotation around the axis of the toroid, see the figure displayed in the card.

Max edge length, φ direction: The maximum edge length along the curved edge in the φ direction in m (is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.

Max edge length, α direction: The maximum edge length along the curved edge in the α direction in m (is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.

Normal vector directed: The triangles can be created such that the normal vectors point *Outward* (outward, away the ring axis of the toroid) or *Inward*.

Scale second half axis: If this parameter is empty or is set to 1, a toroid with a circular cross section is created. If set to $\frac{b}{a}$, an elliptical toroid is created by distorting the entire geometry along the second half axis

(orthogonal to the axis S1–S3) with the factor $\frac{b}{a}$ where a is the distance S1–S3. It is recommended to generate toroids where the elliptical cross section has extremely small or extremely large axial ratios with a CAD system (such as FEMAP) as the distortion formulation used in PREFEKO may fail in these cases.

A complete toroid is obtained by using the parameters $\varphi = 360°$ and $\alpha = 360°$.

Examples of TO card usage

The toroidal segment, which is shown in figure 9-40, is generated using a TO card. This card can also be used to generate the toroidal segment with an elliptical cross section as shown in figure 9-41.

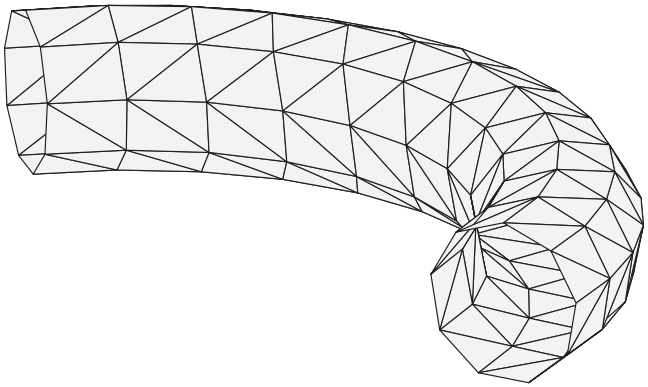


Figure 9-40: Example for the TO card from `demo_TO1.pre`

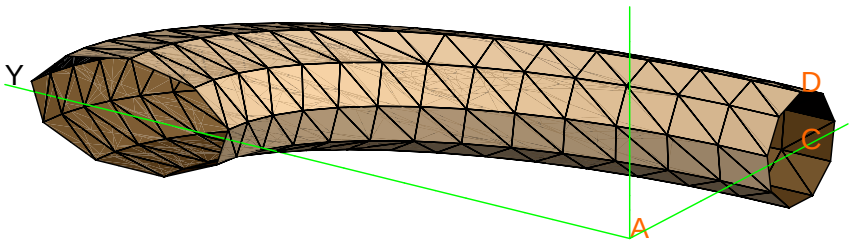
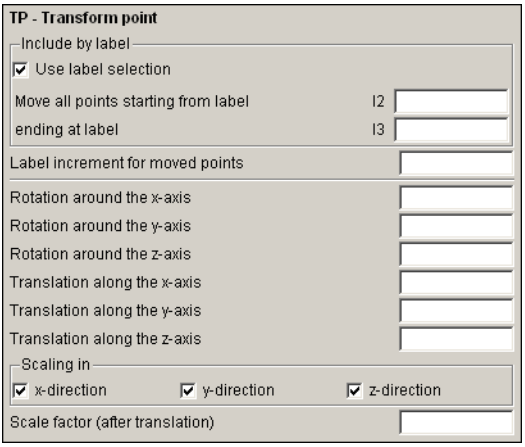


Figure 9-41: Example for the TO card with an elliptical cross section from `demo_TO2.pre`. Note that it is stretched in the direction of the y axis, i.e. it is elliptical in the φ-plane. It is also elliptical in the α-plane when $\varphi = 90°$, but it is circular in the α-plane when $\varphi = 0°$.

9.2.40 TP card

With this card points (previously defined with the DP card) can be translated, rotated and/or scaled (relative to the origin).



Parameters:

Use label selection: If this option is not checked, then the TG card applies to all the previously defined points. If this option is checked, then a label selective processing is possible.

Move all points starting from label: together with *ending at label* specify the label range of points that must be translated, rotated or scaled. See the general discussion of label ranges in section 2.3. If the second field is left empty, only structures with the label set in the first field are considered.

Label increment for moved points: Each transformed point will be assigned a label that is the label of the original point incremented by this value. The exception are points with label 0 — their label is not incremented, it remains 0.

Rotation around the x-axis: Angle of rotation α_x around the x axis in degrees.

Rotation around the y-axis: Angle of rotation α_y around the y axis in degrees.

Rotation around the z-axis: Angle of rotation α_z around the z axis in degrees.

Translation along the x-axis: Translation Δ_x in the x direction. (All three translation distances are affected by the scaling factor set with the SF card.)

Translation along the y-axis: Translation Δ_y in the y direction.

Translation along the z-axis: Translation Δ_z in the z direction.

Scaling in: In this group the user may choose whether scaling is in the x-direction, y-direction or z-direction or a combination of these.

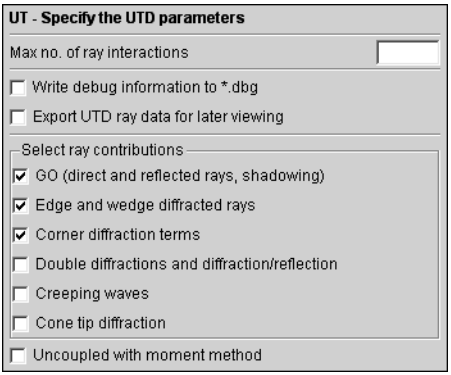
Scale factor: The scaling factor γ, with which the point is scaled after rotation and translation (if the parameter R_7 is not specified, it defaults to $\gamma = 1$).

If a point is rotated around more than one axis with a single card, it is rotated first by an angle α_z around z axis, then by α_y around the y axis and finally by α_x around the x axis. A more detailed description of the transformation can be found in the description of the TG card (section 9.2.38).

In an exception to the rule that all geometry cards must appear before the EG card, this card (as well as the DP card) can be used after the EG to define points for use in the AP card.

9.2.41 UT card

With this command the parameters for the Uniform geometric theory of diffraction (UTD) for the polygonal surfaces are defined.


UT - Specify the UTD parameters
Max no. of ray interactions
Write debug information to *.dbg
Export UTD ray data for later viewing
Select ray contributions
GO (direct and reflected rays, shadowing)
Edge and wedge diffracted rays
Corner diffraction terms
Double diffractions and diffraction/reflection
Creeping waves
Cone tip diffraction
Uncoupled with moment method

Parameters:

Max no. of ray interactions: This parameter gives the maximal number of ray-interactions (i.e. reflection and diffraction combined). If, for example, the parameter is set to 3, a ray can have 3 reflections, or 2 reflections and a diffraction. If set to 0, only direct rays are taken into account.

*Write debug information to *.dbg:* If this item is checked a debug file (extension `*.dbg`) is generated. This file contains large amounts of information and should only be used when debugging.

Export UTD ray data for later viewing: When this item is checked the ray information is exported to the `*.bof` and to a special `*.ray` file, so that the ray paths can be displayed in POSTFEKO. The ray information can become very large, and thus it should only be exported if specific ray paths are to be examined.

Select ray contributions: Determines which ray contributions to take into account:

- *GO (direct and reflected rays, shadowing):* Geometric optics (GO), i.e. direct and reflected rays and shadow regions are taken into account.
- *Edge and wedge diffracted rays:* Diffraction on edges and wedges are taken into account. The ray may include an arbitrary number of reflections, but only one diffraction. (The total number of interactions — the number of reflections plus one for the diffraction — must not be larger than specified in the *Max no. of ray interactions* field.)

- *Corner diffraction terms:* Corner diffraction.
- *Double diffractions and diffraction/reflection:* Double diffraction on edges and wedges and combinations of reflections are taken into account. Single diffraction rays are not included in this item.
- *Creeping waves:* Creeping waves on curved surfaces.
- *Cone tip diffraction:* Tip diffraction at the tip of a cone.

Uncoupled with moment method: This item specifies whether the coupling from the UTD region to the MoM region should be considered. This option should only be used when the UTD and MoM regions are not close together.

When no UT card is used, the following default values apply:

- *Max no. of ray interactions:* 3
- *Write debug information to *.dbg:* unchecked
- *Export UTD ray data for later viewing:* unchecked
- *Select ray contributions* includes
  - *GO (direct and reflected rays, shadowing)*
  - *Edge and wedge diffracted rays*
  - *Corner diffraction terms*

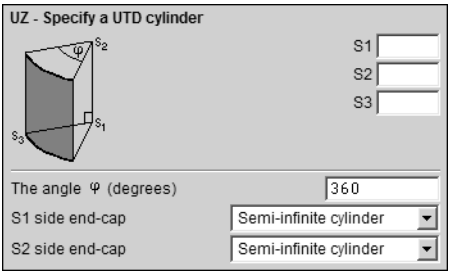
Increasing the type and number of ray interactions increases accuracy but computation time as well. The user should therefore make a compromise between the number of ray interactions and the ray contributions. Choices made in this card should be made on physical considerations to get optimal use from the UTD formulation.

The following restrictions apply for the hybrid MoM/UTD:

- no dielectric bodies or dielectric ground
- only perfectly conducting flat polygonal plates or a single cylinder allowed in the UTD region
- no UTD and PO at the same time

9.2.42 UZ card

With this card a cylinder is created for the UTD region.



Parameters:

S1:	The start point of the cylinder axis.
S2:	The end point of the cylinder axis.
S3:	A point on the radius of the cylinder, the angle S2–S1–S3 must be 90°.
The angle φ:	The angle of the cylinder segment. Currently this must be 360°.
S1 side end-cap:	Select a flat end cap or a semi-infinite end on the side of S1.
S2 side end-cap:	Select a flat end cap or a semi-infinite end on the side of S2.

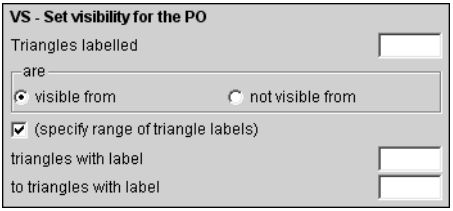
Example of UZ card usage:

The cylinder in figure 9-42 was created using a UZ card. Note the absence of discretisation.

Figure 9-42: Example for UZ card from `demo_UZ1.pre`

9.2.43 VS card

This card specifies known visibility information (required when using physical optics with multiple reflections) to reduce the time required to calculate it.



To accurately compute multiple reflections the code needs to determine which basis functions are visible to each other. Since this applies to all the PO triangles it may be very time consuming for large problems. The time required to determine the visibility may be greatly reduced if the user can inform the code that certain triangles are hidden from each other and others are visible to each other.

Parameters:

Triangles labelled:	The label of the source triangles.
are visible from:	All triangles with the label specified in the field *Triangles labelled* are visible from all triangles with label(s) indicated in the fields below.
are not visible from:	All triangles with the label specified in the field *Triangles labelled* are not visible from all triangles with label(s) indicated in the fields below.
(specify range of labels):	If this item is unchecked, only a single label is specified (in the field *triangles with label*). If checked, the card applies to all triangles with labels in the range from the value specified in the *triangles with label* field to that in the *to triangles with label* field.

Note that visibility is reciprocal, i.e. if all triangles with label n are visible from all triangles with label m, all triangles with label m are visible from all triangles with label n as well.

Basis functions cannot illuminate each other if all the triangles they are attached to lie in the same plane.

The VS card should only be used if the user can specify the visibility beyond any doubt and if it applies to all triangles of that label. If no information is specified for a specific combination of labels/triangles, full ray tracing will be executed.

Example of VS card usage:

Consider the structure shown in figure 9-43 consisting of four flat plates and a cylindrical section. The two plates lying at 45 degrees to the coordinate system (labelled 1 and 3), are half as wide as the plates with labels 0 and 2. Thus some triangles with label 2 are visible to some triangles with label 0, but not all.

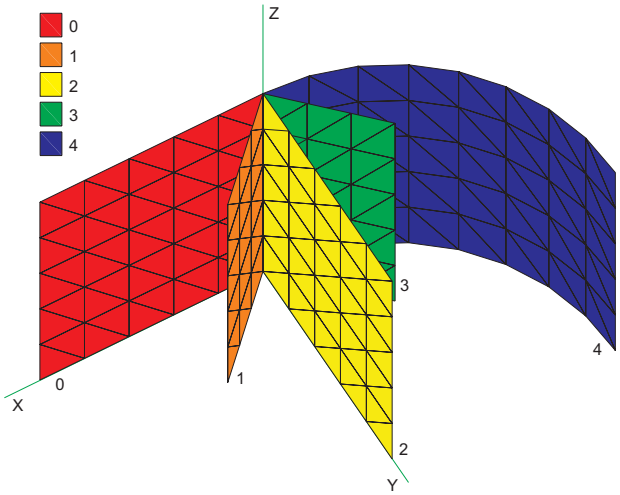


Figure 9-43: Structure used to demonstrate the use of VS cards from `demo_VS1.pre`

We have to specify which triangles are visible\hidden from *all* triangles with label 0 first, then those visible from label 1 and so on. The VS cards for this example would be as follows:

- *Triangles labelled* 0 *are not visible from triangles with label* 0.
- *Triangles labelled* 0 *are visible from triangles with label* 1.
- *Triangles labelled* 0 *are not visible from triangles with labels* 3 *to* 4.
- *Triangles labelled* 1 *are not visible from triangles with label* 1.
- *Triangles labelled* 1 *are visible from triangles with label* 2.
- *Triangles labelled* 1 *are not visible from triangles with labels* 3 *to* 4.
- *Triangles labelled* 2 *are not visible from triangles with label* 2.
- *Triangles labelled* 2 *are visible from triangles with label* 3.
- *Triangles labelled* 3 *are not visible from triangles with label* 3.
- *Triangles labelled* 3 *are visible from triangles with label* 4.

Since all the triangles with label 0 lie in the same plane, they cannot illuminate each other. Thus the first card states that label 0 is hidden from label 0.

All triangles with label 1 are visible from all triangles with label 0. This is specified by the second VS card. Since some triangles with label 2 are visible from some triangles with label 0 while others are hidden, we cannot specify any information for this combination of layers. However, the plate with label 2 shadows all triangles with labels 3 and 4 and we may specify that these are hidden. This is done with the third VS card. Note that this card specifies a range of hidden labels.

Next we must specify which triangles are visible (or hidden) from *all* triangles with label 1. As for label 0, triangles with label 1 are not visible to each other — specified by the fourth VS card. All triangles with labels 0 and 2 are visible from all triangles with label 1. Since we have already specified the visibility between labels 0 and 1, we do not specify it again. The fifth VS card then specifies that label 2 is completely visible from label 1. As for label 0, both labels 3 and 4 are hidden completely which completes the first six VS cards,

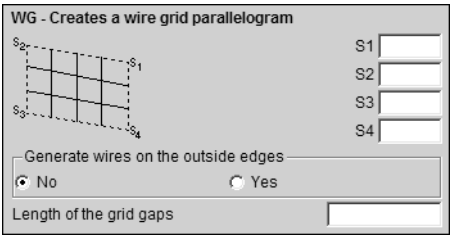
Next we look at label 2. As before we need not consider labels lower than 2. Also the label is hidden from itself as indicated by VS card number seven. Next we state that label 3 is visible, but we cannot specify anything about label 4 as only some of these triangles will be visible.

Similarly VS cards 9 and 10 states that label 3 is not visible to itself and fully visible to label 4.

Finally we must consider the case for triangles with label 4. All visibility with layers 0 to 3 has been specified and may not be specified again. Unlike the previous flat plates, layer 4 is curved and some triangles may indeed illuminate other triangles with the same layer. However, not all other triangles will be illuminated (this is only possible for a doubly concave surface), such that we cannot specify any information for label 4.

9.2.44 WG card

With this card a wire grid in the shape of a parallelogram can be generated.



Parameters:

S1, S2, S3, S4: The four corners of the parallelogram in consecutive order.

Generate wires on the outside edges: If the *No* item is selected, only the wires inside the parallelogram are generated whereas if the *Yes* item is selected all the wires are generated. This option is important when two adjacent parallelograms are generated, as the segments along the sides must not be generated twice.

Length of the grid gaps: The maximum segment length is given by the IP card. This parameter is an integer number and specifies the density of the grid. If, for example, this is set to 2 the wires only cross at every second segment.

Example of WG card usage:

The wire grid seen in figure 9-44 is generated using the WG card.

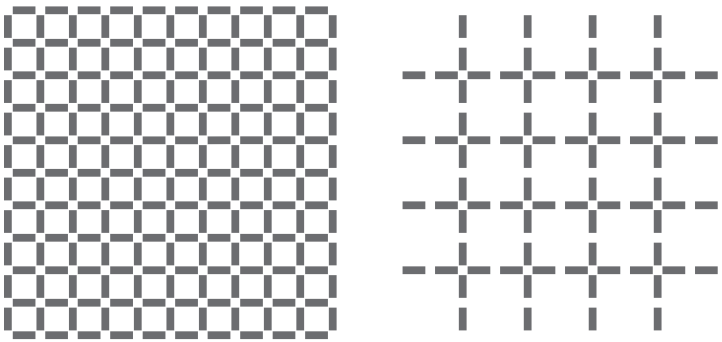
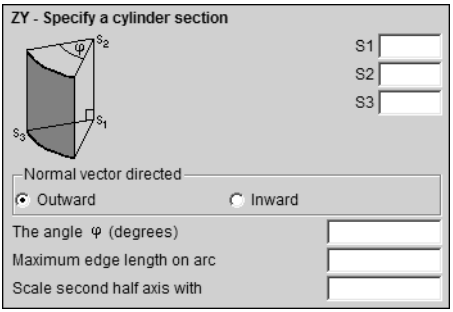


Figure 9-44: Examples of the WG card from `demo_WG1.pre` — the one on the right has a value of 2 in *Length of the grid gaps*.

9.2.45 ZY card

With this card a surface mesh in the form of a cylindrical segment can be generated.



Parameters:

S1:	The start point of the axis.
S2:	The end point of the axis.
S3:	A point on the corner of the cylindrical segment.
Normal vector directed:	The triangles can be created such that the normal vector is points *Outward* or *Inward*.
The angle φ:	The angle in degrees, which is subtended by the cylindrical arc.
Maximum edge length on arc:	Maximum edge length of the triangles along the curved side in m (is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.
Scale second half axis:	If this parameter is empty or is set to 1, a circular cylinder is created. If set to $\frac{b}{a}$, an elliptical cylinder is created. Here $\frac{b}{a}$ gives the ratio of the two half axes, where a is the distance S1–S3. It is recommended to generate elliptical cylinder with extremely small or extremely large axial ratios with a CAD system as the distortion formulation used in PREFEKO may fail in these cases.

For an orthogonal cylinder (i.e. the lines S1–S2 and S1–S3 are perpendicular), the segmented area (shaded in the figure of the cylinder) is obtained by rotating the point S3 around the axis S1–S2 through φ. For $\varphi = 360°$ a full cylinder is obtained.

An oblique cylinder (i.e. the circular or elliptical rim is not perpendicular to the axis) can also be created. Then S1–S2 still represents the axis, but the top and bottom planes of the rims are defined by planes perpendicular to the plane defined by the three points S1, S2, S3, and parallel to the line S1–S3.

Examples of ZY card usage:

The cylindrical section (figure 9-45), elliptical cylinder (figure 9-46) and non-orthogonal cylinder (figure 9-47) below were all created with ZY cards.

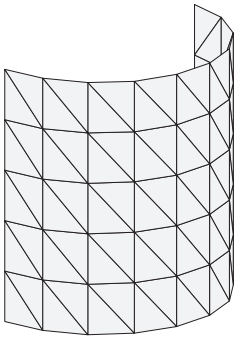


Figure 9-45: Example for the ZY card from `demo_ZY1.pre`

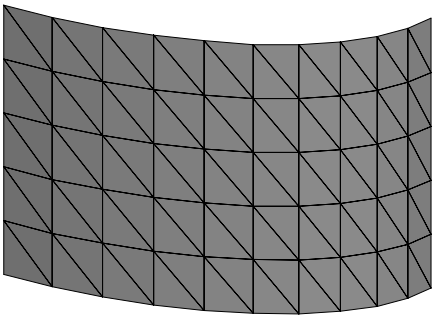


Figure 9-46: Example of an elliptical cylinder from `demo_ZY2.pre`

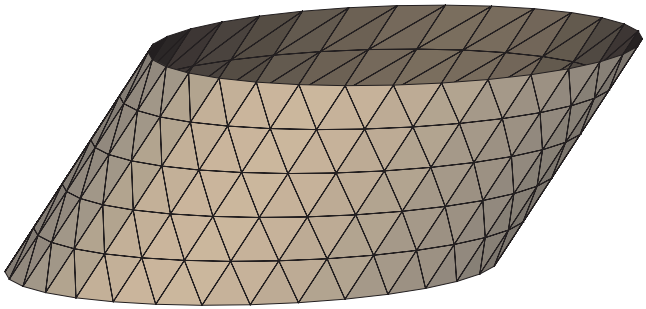


Figure 9-47: Example of a non-orthogonal elliptical cylinder from `demo_ZY3.pre`

10 Description of the control cards

10.1 Overview of control cards and execution sequence

The following table summarises all the PREFEKO and FEKO control cards. These cards should not be used in the geometry section of the `*.pre` input file, i.e. they should only be used below the EG card. The control cards are used to specify, for example, the frequency and the type of excitation. They also determine the required calculations (such as the locations for near field calculations and the directions for far field calculations, etc.).

Card	Description
**	characters used to indicate a comment line
Ax	type of excitation (e.g. an incident plane wave or a voltage source)
BO	through the use of the reflection factor coefficient a ground plane can be inserted
CF	set the type of integral equation for perfectly conducting metallic surfaces
CG	the algorithm used to solve the matrix equation is selected
CM	Field calculation for CableMod (coupling into transmission lines) or PCBMod (couling into a PCB)
CO	inserts a dielectric and/or magnetic surface on the elements
DA	creates additional files for the results
DI	enters the properties of the dielectric medium
EN	indicates the end of the input file
FE	calculates the near fields
FF	calculates the far fields
FR	sets the frequencies at which the calculations are to be carried out
GF	sets the Green's functions
L4	adds a load between a metallic triangle and the ground plane for the planar multilayer Greens function
LD	defines a distributed load, consisting of resistance, inductance and capacitance
LE	defines a load on the edge between surface triangles
LP	defines a parallel circuit (resistance, inductance and capacitance) load
LS	defines a series circuit (resistance, inductance and capacitance) load
LZ	defines a complex load
OF	offset i.e. displacement of the origin when calculating the near fields
OS	saves the surface currents in a file
PS	sets general control parameters
PW	defines the radiating power of a transmitting antenna
SA	Used to calculate SAR in dielectric media
SK	takes a finite conductivity into account through the skin effect of ohmic losses; also for thin dielectric layers

SP	calculates the S-parameters for the active sources
TL	specifies a non-radiating transmission line

As mentioned above the control cards form the second part of the input file (see also section 2.1). Control cards are processed line by line and only affect other cards and calculations specified below them in the input file. (Information specified in a control card is not available before that line is processed.) Any number of control cards can be used, but they should adhere to a basic sequence. Thus, for example, the frequency (FR card) and the type of excitation (Ax card) must be defined before the near fields can be calculated with an FE card.

In addition, a sensible order for the control cards can result in a considerable reduction in the computation time. For example, for an SK card the whole matrix has to be recalculated, while an Ax card only redefines the right hand side of the matrix. A summary of the dependencies is given in the table below:

Action	For the cards
recalculates matrix elements	BO, CF, CO, DI, FR, GF, L4, LD, LE, LP, LS, LZ, SK, SP, TL
recalculates the right hand side	Ax, BO, DI, FR, GF
resolves the matrix equation	CG

There are also other dependencies. If the matrix elements are recomputed then the matrix equation has to be solved again. The actual calculation is started by the CM, FE, FF, OS, SA and SP cards. All other cards are read and the data stored.

When solving for a number of frequencies all the control cards following the FR card (up to, but excluding, the next FR card or EN card) are read into a buffer. For each frequency all these cards are read and processed. The computation time can be reduced significantly by using the cards in the correct order. If, for example, a structure needs to be investigated at two frequencies and with two different excitations then the control cards can be organised in either of the following ways:

```
...
FR   FR card for the two frequencies
Ax   first excitation
...
Ax   second excitation
...
EN   end of the input file
```

or

```
...
FR   FR card for the two frequencies
Ax   first excitation
...
FR   FR card for the two frequencies
Ax   second excitation
...
EN   end of the input file
```

In the first format, the two excitations are processed one after the other for each frequency. (The cards are executed in the order FR Ax ... Ax ... FR Ax ... Ax ... EN, where the second FR indicates execution of the frequency loop for the second specified frequency.) In the second format the first excitation is treated for both frequencies before treating the second excitation for both frequencies. (Here the cards are executed as FR Ax ... FR Ax ... FR Ax ... FR Ax ... EN.) For the computational requirements, one finds that in the first case the matrix elements have to be calculated twice (it have to be completely recalculated each time the frequency is changed) and in the second case four times.

The computation time is not only influenced by the control cards, but also by what has to be solved for. In the following example, the structure has to be solved at a number of frequencies and for the ideal (conducting) and non-ideal (conduction with losses) cases:

```
...
FR   FR card for multiple frequencies
FE   calculation of the near fields
SK   include skin effect
FE   calculation of the near fields
EN   end of the input file
```

The three control cards FE, SK and FE are written to the buffer and are worked through in the loop for the different frequencies. At the first frequency the FE card initiates the field calculation. Because a SK card has not yet been read, ideal conductivity is assumed. Then the SK card is read and losses are taken into account when the second FE card is run. Thus the first frequency pass is finished. At the next frequency pass the cards FE, SK, and FE are read again, but the losses from the SK card are still active from the first pass. The SK card is thus useless and the two FE cards calculate the same things twice. The correct input order is:

```
...
FR   FR card for multiple frequencies
SK   skin effect switched off
FE   calculation of the near fields
SK   skin effect switched on
FE   calculation of the near fields
EN   end of the input file
```

Then the four cards SK, FE, SK and FE are calculated for all the frequencies.

10.2 Detailed description of the control cards

10.2.1 ** card

The comment lines discussed in section 9.2.1 can also be used after the EG card.

10.2.2 Ax Cards

This card defines the type of excitation as well as other parameters regarding the excitation. The following possibilities are available:

Card	Type of Excitation
A0	A linear polarised plane wave incident on the structure.
A1	Excitation by means of a voltage gap on a segment (i.e. impressed electric field strength along a segment).
A2	Excitation by means of a voltage gap at a node i.e. between two segments.
A3	Excitation by means of a magnetic ring current (TEM-Frill) on a segment. Thus a coaxial feed can be modelled.
A4	Special vertical pin excitation, e.g. for a patch antenna on a planar substrate with a ground plane (coaxial probe excitation mode).
A5	A Hertzian dipole is used as excitation. The position and orientation in the space are arbitrary.
A6	A Magnetic dipole is used as excitation. The position and orientation in the space are arbitrary.
A7	Excitation by means of a voltage gap on an edge between two triangles. This card has been generally replaced by the AE card.
AC	This card reads the geometry and current distribution (possibly for more than one frequency) from an `*.rsd` file created by the transmission line simulation program CableMod[7] or by the PCB simulation tool PCBMod or by export with the OS card in FEKO. The excitation is due to the electromagnetic fields radiated by these line currents.
AE	The AE card is an excitation between triangle edges similar to the A7 card, however the AE card permits the simultaneous excitation of several edges.
AI	Excitation by an impressed line current.
AP	Excitation with an aperture (array of electrical and magnetic Hertzian dipoles).
AR	Excitation by an antenna with a given radiation pattern.
AS	Excitation by means of incident of radiating spherical modes.
AV	Excitation by an impressed line current similar to the AI card, but the endpoint of the current is electrically connected to a conducting surface.
AW	Excitation by an impressed mode on a waveguide port.

[7] To use this the CableMod/PCBMod module must be activated, if required please contact your distributor.

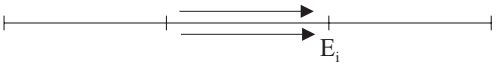
The different ways to realise a voltage source are summarised in figures 10-1 and 10-2. The impressed electric field strength is indicated by $\vec{E}_i$.

More than one excitation is also allowed. Thus one may, for example, generate an elliptically polarised plane wave by super-imposing two out of phase linearly polarised plane waves with different amplitudes. It is also possible to feed an antenna with two different voltage sources. For this purpose the parameters *New source* and *Add to sources* are available in each Ax card. This parameter indicates whether the current excitation is additional (*Add to sources*) or not (*New source*). When *New source* is selected, only the current excitation will be used and the excitations prior to the current one will be erased.

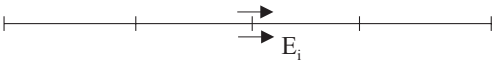
For the excitations A1, A2, A3, A4 and A7 it is possible to select the feed element through the label. Alternatively the position of the feed is specified in Cartesian coordinates. This should, simplify the modelling in most cases — especially with the A7 card. FEKO then searches for a segment or an element at this position. This comparison of the position uses the tolerance parameter *Maximum identical distance* (EG card, section 9.2.12).

The excitations are described in detail in the following sections.

A1 card: Voltage source on a segment



A2 card: Voltage source on a node between segments



A3 card: TEM-frill on a segment

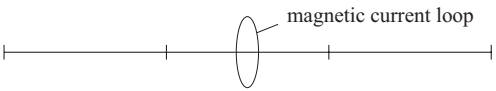
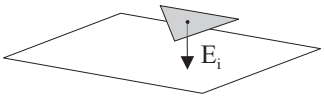
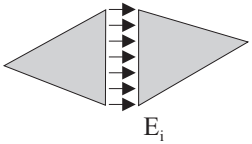


Figure 10-1: Possible ways to realise a voltage source on a wire segment

A4 card: Vertical pin approximation (dielectric substrate)



A7 card: Voltage gap on an edge



AE card: Voltage gap along edges

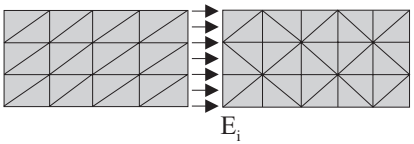
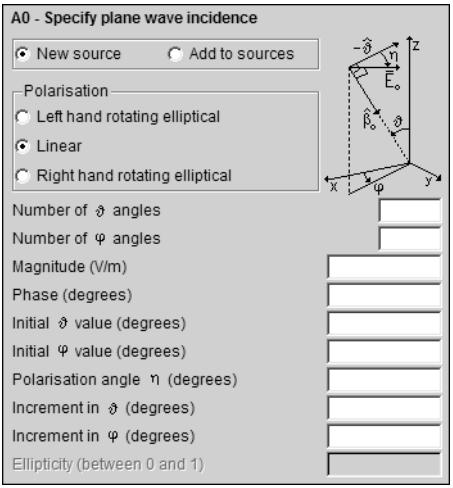


Figure 10-2: Possible ways to realise a voltage source in connection with triangles

10.2.3 A0 card

This card realises excitation by a linearly polarised incident plane wave.



Parameters:

New source:	New excitation, replaces all previous excitations.
Add to sources:	Additional excitation, is added to previous excitations.
Polarisation:	This group sets the behaviour of the polarisation vector. If either of the rotating options is selected, the *Ellipticity* must be specified below.
Number of ϑ angles:	If more than one direction of incidence is to be examined, then this parameter indicates the number of incident angles in the ϑ direction. If this field is left empty or set to 0, a value of 1 will be used.
Number of φ angles:	If more than one direction of incidence is to be examined, then this parameter indicates the number of incident angles in the φ direction. If this field is left empty or set to 0, a value of 1 will be used.
Magnitude:	Magnitude of the field strength $\vec{E}_0$ of the incident field in $\frac{\mathrm{V}}{\mathrm{m}}$.
Phase:	Phase of the field strength $\vec{E}_0$ of the incident field in degrees.
Initial ϑ value:	Angle of incidence ϑ of the plane wave in degrees, see the figure in the card above.

Initial φ value:	Angle of incidence φ of the plane wave in degrees, see the figure in the card above.
Polarisation angle η:	See the figure in the card above. It is the angle — in a right-handed sense when viewing in the incident direction — from $-\hat{\vartheta}$ to $\vec{E}_0$.).
Increment in ϑ:	If more than one direction of incidence is to be examined, ϑ is incremented by this value for each new angle of incidence.
Increment in φ:	If more than one direction of incidence is to be examined, φ is incremented by this value for each new angle of incidence.
Ellipticity:	This field is only applicable when elliptical rotation is selected under *Polarisation* above and gives the ellipticity of the rotating polarisation. It must be larger than 0 (linear polarisation) and smaller or equal to 1 (circular polarisation).

The direction of incidence $\hat{\beta}_0$ is specified by the incidence angles ϑ and φ. The polarisation angle η (measured from the negative of the spherical coordinate system unit vector $\hat{\vartheta}$) and the field strength vector $\vec{E}_0$ is defined as indicated in the figure in the card above.

The electric field strength of the incident field is then given by

$$\vec{E}_i(\vec{r}) = \left[\vec{E}_0 + jv(\vec{E}_0 \times \hat{\beta}_0)\right] \cdot e^{-j\vec{\beta}_0\vec{r}}$$

where v is the ellipticity. For linear polarisation it is 0; for right hand rotating it is equal to the value in the *Ellipticity* field; for left hand rotating it is the negative of the value in the *Ellipticity* field. The incident magnetic field is given by

$$\vec{H}_i(\vec{r}) = \frac{1}{Z_F}\,\hat{\beta}_0 \times \vec{E}_i$$

with Z_F the wave impedance in the surrounding free space medium.

It should be noted that the incident power density (which is required, for example, for RCS computations) is given by

$$\vec{S}_i = \frac{1}{2}\,\frac{|\vec{E}_0|^2}{Z_F}\,(1+v^2)\ .$$

It is possible to vary the direction of incidence. This is particularly useful when e.g. determining the monostatic radar cross section. The two parameters *Initial ϑ value* and *Initial φ value* indicate the direction of the first wave. The direction of incidence is varied in the ϑ direction by the increment *Increment in ϑ* and in the φ direction by *Increment in φ*. In each direction these two angles are examined and a total number of incident waves equal to the product of these two angles are examined.

If an A0 card with either *Number of ϑ angles* or *Number of φ angles* larger than 1 is read, all the following control cards up to, but excluding, the next Ax, FR or EN card will be read into a buffer. All these cards are then processed in a loop, over all the different angles of incidence. If e.g. the monostatic radar cross section is to be calculated for $\vartheta = 90°$ and $0° \leq \varphi \leq 180°$, the following command is used:

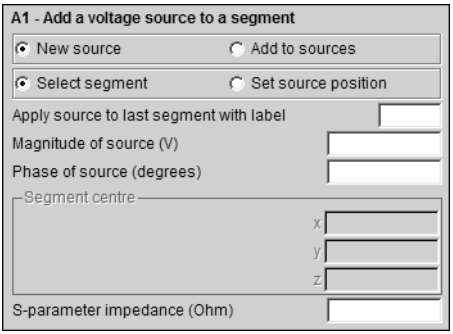
```
...
A0   0     1 181     1.0    0.0   90.0    0.0    0.0    0.0    0.1
FF   2
EN
```

In this demonstration file, the FF card is read into the buffer and processed 181 times. Through the use of the parameter *Fields calculated only in incident direction* in the FF card the far field is calculated in the direction of the incident wave.

If more than one direction of incidence is to be examined, the right hand side of the linear equation system is changed, but the matrix remains unchanged. Thus it makes sense, by using the CG card, to use Gauss elimination (default if a CG card is not used) which performs a LU-decomposition of the matrix. When the direction of incidence is varied, then only the relatively fast backward substitution has to be done.

10.2.4 A1 card

With this card a voltage source is placed on a segment.



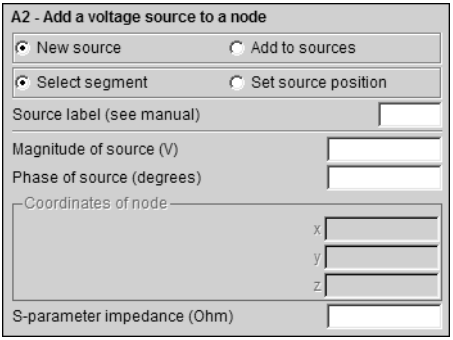
Parameters:

New source:	New excitation, replaces all previous excitations.
Add to sources:	Additional excitation, is added to previous excitations.
Select segment:	When this item is selected, the *Apply source to last segment with label* field becomes active. Here one specifies the label of the segment to which the source must be applied. If more than one segment has this label, the source is applied to the last segment with this label. Alternatively, one may select the item *Set source position,* then the feed segment is determined by specifying the Cartesian coordinates in the *Segment centre* group. These values are in m and are scaled by the SF card if *Modify all dimension related values* is checked.
Magnitude of source:	Magnitude of the voltage $\|U_0\|$ in V.
Phase of source:	Phase of the voltage U_0 in degrees.
S-parameter impedance:	The port impedance if this excitation is used in connection with S-parameter calculation. If this field is empty or 0, the value specified at the SP card is used. (This value is only used if the S-parameters are requested with an SP card.)

The vector of the voltage lies in the direction from the beginning of the segment to its end, in the direction in which the segment was created by the BL card. (This is the direction of the current flow through the segment. The internal EMF — electromagnetic force — of the impressed voltage source is in the opposite direction.)

10.2.5 A2 card

With this card a voltage source is placed at a node between two segments or between a segment and a triangle, ground plane or polygonal plate. It is mostly used to feed wires attached to plates, etc.



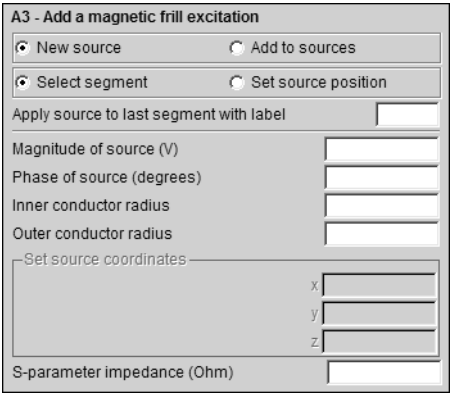
Parameters:

New source:	New excitation, replaces all previous excitations.
Add to sources:	Additional excitation, is added to previous excitations.
Select segment:	When this item is selected, the *Source label* field becomes active. Here one specifies the label of the segment at which start point the source is located. The excitation has to be located at a node, either between two segments, or between a segment and a triangle, ground plane or polygonal plate. Only one segment with this label should be declared. If there is more than one segment with this label then only one node will be fed. Alternatively, one may select the item *Set source position,* then the feed node is determined by specifying its Cartesian coordinates in the *Coordinates of node* group. These values are in m and may be scaled by the SF card.
Magnitude of source:	Magnitude of the voltage $\|U_0\|$ in V.
Phase of source:	Phase of the voltage U_0 in degrees.
S-parameter impedance:	The port impedance if this excitation is used in connection with S-parameter calculation. If this field is empty or 0, the value specified at the SP card is used. (This value is only used if the S-parameters are requested with an SP card.)

There may not be more than two segments connected to the node. The direction of the vector points in the same direction as the basis function that has been assigned to this node. (When only one segment is connected to the node, the direction is away from the segment. This is the direction of the current flow through the node. The internal EMF — electromagnetic force — of the impressed voltage is in the opposite direction.)

10.2.6 A3 card

This card realises excitation by a magnetic ring current (TEM-frill) on a segment. This will give an accurate model of a coaxial feed, but requires both the inner and outer radii.



Parameters:

New source:	New excitation, replaces all previous excitations.
Add to sources:	Additional excitation, is added to previous excitations.
Select segment:	When this item is selected, the *Apply source to last segment with label* field becomes active. Here one specifies the label of the segment on which the TEM-frill is placed. If more than one segment has this label, the source is applied to the last segment with this label. Alternatively, one may select the item *Set source position,* then the feed segment is determined by specifying the Cartesian coordinates in the *Segment centre* group. The position values are in m and are scaled by the SF card if *Modify all dimension related values* is checked.
Magnitude of source:	Magnitude of the voltage $\|U_0\|$ in V.
Phase of source:	Phase of the voltage U_0 in degrees.
Inner conductor radius:	Radius of the inner conductor of the coaxial feed.
Outer conductor radius:	Radius of the outer conductor of the coaxial feed.
S-parameter impedance:	The port impedance if this excitation is used in an S-parameter calculation. If this field is empty or 0, the value specified at the SP card is used. (This value is ignored if no SP card is used.)

The vector of the excitation points from the start point to the end point of the segment, i.e. in the direction in which the segment was created with the BL card. The excitation is not, as in the A2 card, an impressed electric field strength, but is a magnetic ring current.

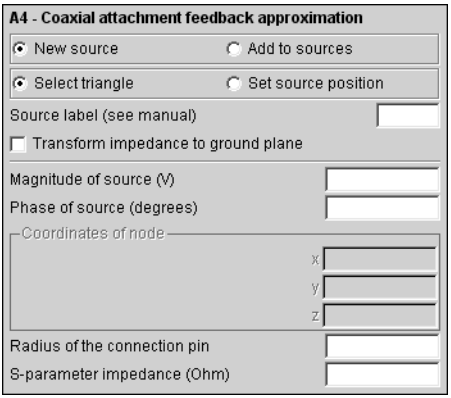
As a rule of thumb, the radius of the inner conductor must be the same as the radius of the segment and that the outer radius should be 2 to 3 times the size of the inner. If an impedance Z is desired, then the following relation can be used

$$Z \approx 60\,\Omega \cdot ln\,\frac{Outer\ conductor\ radius}{Inner\ conductor\ radius}$$

to determine the outer radius. For $Z = 50\,\Omega$ the outer conductor radius should be equal to 2.3 times the inner conductor radius.

10.2.7 A4 card

This card creates a coaxial attachment feed approximation for use in connection with the Green's function for planar substrates with a metallic ground plane (GF card, section 10.2.30).



Parameters:

New source: New excitation, replaces all previous excitations.

Add to sources: Additional excitation, is added to previous excitations.

Select triangle: When this item is selected, the *Source label* field becomes active. Here one specifies the label of the triangle to excite. The feed point is at the centroid of the triangle (see figure 10-3). If there are more than one triangle with this label, the excitation is placed on the one with the highest element number. Alternatively, the user may select the item *Set source position,* and specify the Cartesian coordinates of the feed point in the *Coordinates of node,* group. (These values are in m and optionally scaled by the SF card.) FEKO will excite the triangle whose centroid is closest to the specified coordinate.

Transform impedance to ground plane: If unchecked the impedance is computed directly at the excitation point. If checked, an inductive approximation of the feed pin is used and the impedance is transformed to the ground plane.

Magnitude of source: Magnitude of the voltage $|U_0|$ in V.

Phase of source: Phase of the voltage U_0 in degrees.

Radius of the connection pin: The radius of the coaxial probe feed pin in m. This value is optionally scaled by the SF card.

S-parameter impedance: The port impedance if this excitation is used in connection with S-parameter calculation. If this field is empty or 0, the value specified at the SP card is used. (This value is only used if the S-parameters are requested with an SP card.)

The excitation is shown in figure 10-3. A typical application of the A4 card is given in `example_30a.pre` (*Examples Guide*). It is of course possible to discretise the vertical pin (into segments) and feed one of the segments with a voltage source (A1 card). The advantage of the A4 card is that there are no vertical currents, which results in substantially simpler Green's functions and a significant reduction in computing time.

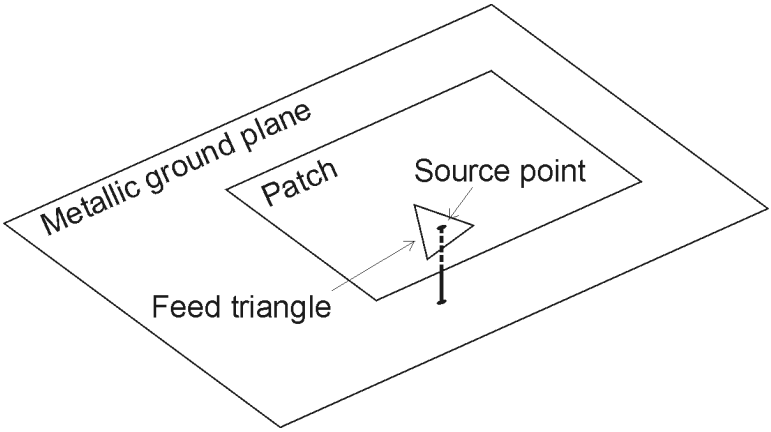
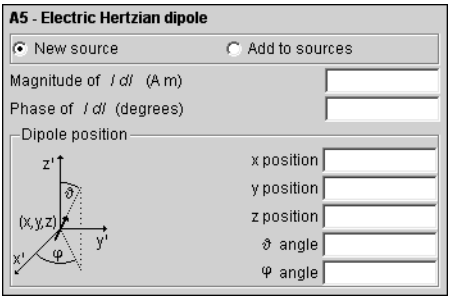


Figure 10-3: Excitation of a patch antenna with a vertical pin

10.2.8 A5 card

This card specifies excitation by an electric Hertzian dipole.



Parameters:

New source:	New excitation, replaces all previous excitations.
Add to sources:	Additional excitation, is added to previous excitations.
Magnitude of I dl:	Absolute value of the complex amplitude $I \cdot l$ in Am.
Phase of I dl:	Phase of the complex amplitude $I \cdot l$ in degrees.
x, y, z position:	Coordinates of the position of the dipole in m. These values are optionally scaled by the SF card.
ϑ angle:	Orientation of the dipole in space — ϑ (in degrees) is the angle between the dipole and the z axis.
φ angle:	Orientation of the dipole in space — φ (in degrees) is the angle between the projection of the dipole onto the plane $z = 0$ and the x axis.

The dipole moment of the electric dipole is given by

$$p = \frac{I\,l}{j\omega}$$

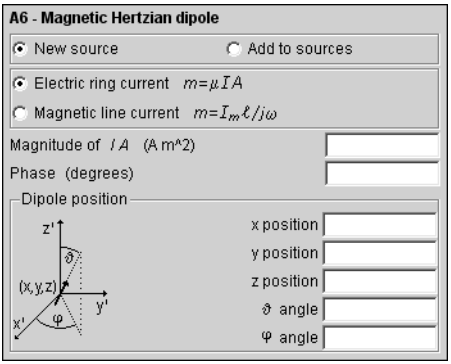
The power radiated by the dipole in a free space environment is given by

$$P = \frac{\beta_0^2 \cdot Z_{F0} \cdot |I\,l|^2}{12\pi} = \frac{\omega^2 \mu_0^2 |I\,l|^2}{12\pi Z_{F0}}$$

FEKO, however, considers the properties of the medium in which the dipole is located as well as the coupling of the dipole with surrounding structures or other sources (for example other Hertzian dipoles in an array antenna) when calculating the power radiated by the Hertzian dipole.

10.2.9 A6 card

This card specifies excitation by an elementary magnetic dipole.



Parameters:

New source:	New excitation, replaces all previous excitations.
Add to sources:	Additional excitation, is added to previous excitations.
Electric ring current:	Use the model of an electric ring current for the magnetic dipole (the moment $m = \mu\, I\, A$ where I is the loop current and A the enclosed surface).
Magnetic line current:	Use the model of a magnetic line current (the moment $m = \frac{I_m\, l}{j\omega}$ where I_m is magnetic line current and l the length of the dipole).
Magnitude of . . . :	The magnitude of the dipole. For the electric ring current it is $I \cdot A$ in Am^2. For the magnetic line current it is $I_m \cdot l$ in Vm.
Phase:	Phase of the complex amplitude in degrees.
x, y, z position:	Coordinates of the position of the dipole in m. These values are optionally scaled by the SF card.
ϑ angle:	The angle between the dipole and the z axis in degrees.
φ angle:	The angle between the projection of the dipole onto the plane $z = 0$ and the x axis in degrees.

The power radiated by the dipole in a free space environment is given by

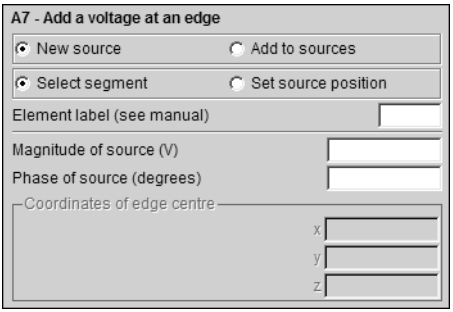
$$P = \frac{\beta_0^4 \cdot Z_{F0} \cdot |I\, A|^2}{12\pi}$$

FEKO, however, considers the properties of the medium in which the dipole is located as well as the coupling of the dipole with surrounding structures or other sources (for example other magnetic dipoles in an aperture approximation — see the AP card) when calculating the power radiated by the Hertzian dipole.

Even though the two formulations, electric ring current and magnetic dipole, result in the same near and far fields, if the dipole moment m is the same, the radiated potentials are different. The electric ring current model gives rise to a magnetic vector potential $\vec{A}$, while the magnetic dipole model results in an electric vector potential $\vec{F}$ as well as a magnetic scalar potential Ψ.

10.2.10 A7 card

This card is used to specify a voltage source on an edge between two triangles or at a connection edge between a single triangle and a PEC ground plane or UTD plate. The AE card is substantially simpler to use and should be used for all new models. (The A7 card is supported only for compatibility with FEKO input files that were created before the AE card became available.)



Parameters:

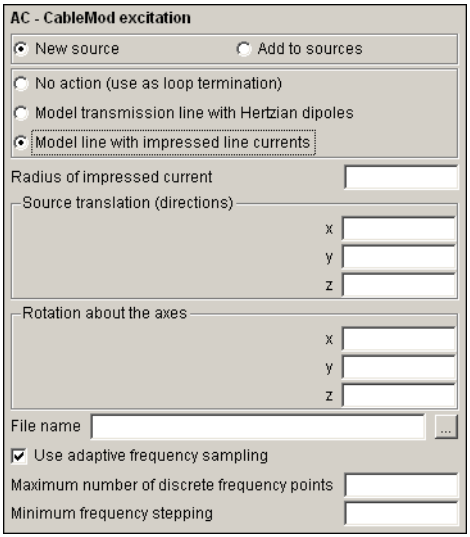
New source:	New excitation, replaces all previous excitations.
Add to sources:	Additional excitation, is added to previous excitations.
Select segment:	If this item is selected, a triangle with label specified in the *Element label* field is searched for. The excitation is placed on the edge that lies opposite to the first corner of the triangle. Once again the label must be unambiguous, i.e. if possible only one triangle must have this label. If there is more than one triangle with this label then only one will be fed. Alternatively, when selecting the item, *Set source position,* the feed edge is determined by specifying its Cartesian coordinates in the *Coordinates of edge centre*. These coordinates are in m and optionally scaled by the SF card. The edge must lie between two triangles or between a triangle and a ground plane or UTD plate.
Magnitude of source:	Absolute value of the voltage $\|U_0\|$ in V.
Phase of source:	Phase of the voltage U_0 in degrees.

If two triangles are connected to the edge, the basis function between these triangles is excited. The vector direction of the voltage source lies in the same direction as the basis function associated with this edge. (This is the direction of the current flow through the edge. The internal EMF (electromagnetic force) of the impressed voltage source is in the opposite direction.)

In certain special cases there may be only one triangle connected to the edge. If the edge lies in the plane of a polygonal UTD plate or a PEC ground plane (specified with a GF or BO card), the excitation is placed on the appropriate basis function connecting the triangle to the plate/plane. The positive feed direction is then towards the edge.

10.2.11 AC card

This card inputs data from a `*.rsd` file containing the geometry of a transmission line or PCB structure and the current distribution along this line or on the PCB for one or more frequencies. Such a `*.rsd` file is created, for example, by the transmission line simulation program CableMod or by the PCB code PCBMod[8] or by a current export with the OS card in FEKO. The excitation is due to the electromagnetic fields radiated by these line currents (the CM card allows the treatment of electromagnetic fields coupling into lines).



Parameters:

New source: New excitation, replaces all previous excitations.

Add to sources: Additional excitation, is added to previous excitations.

No action: No execution, do not read the `*.rsd` file — this option is used to specify the end of the frequency loop, see below.

Model transmission line with Hertzian dipoles: The line geometry, frequency and currents are read from the `*.rsd` file, and the line is modelled with an array of Hertzian dipoles (see the A5 card). The number of dipoles per line segment is specified with the parameter *Number of dipoles per transmission line*. Note that this model is only valid if the line segments do not make electrical contact with any conducting

[8] To use this the CableMod/PCBMod module must be activated, if required please contact your distributor.

surface. (All the segments in the `*.rsd` file must be of the type *intern* and not *loaded*.)

Model line with impressed line currents: The line geometry, frequency and currents are read from the `*.rsd` file, and the line is modelled with a continuous current distribution using one AI card per line segment. (The AI cards are created automatically by PREFEKO when importing the `*.rsd` file.) If a line segment has a *loaded* endpoint it is automatically modelled by an AV card to allow the electrical contact. The radius of the impressed current element can be set in the parameter *Radius of impressed current*. This parameter is optional and is passed on to the AI and AV cards. (See the description in these cards.) If the parameter is zero or empty a current filament approximation is used.

Source translation (directions): When importing transmission line currents from CableMod or PCB currents from PCBMod, then the coordinate system of these programmes as represented in the `*.rsd` file is used and the source is imported at this position in FEKO. Here an offset can be specified which translates the source in x, y, or z direction. Standard FEKO units are used for these offsets (i.e. metres, but scaled accordingly if a factor is set at the SF card). Note that the units as specified in the `*.rsd` file are not applicable here for the translation parameters (only to the import of the data).

Rotation about axes: Like the translation described above, an imported source can here be rotated and thus positioned arbitrarily. The rotation angles are in degree, and the same definition as also used at the AR or TG cards applies (see section 9.2.38 for details).

File name: The name of the `*.rsd` file.

Use adaptive frequency sampling: Only read the minimum and maximum frequency from the `*.rsd` file and obtain a continuous solution in this frequency band using adaptive frequency sampling. If this option is used, only one AC card is permitted in the `*.pre` file (and no FR cards).

Maximum number of discrete frequency points: When using adaptive frequency sampling, the maximum number of sample points can be specified here. See also the discussion on adaptive frequency sampling at the FR card.

Minimum frequency stepping: When using adaptive frequency sampling it could be necessary to specify the minimum allowable frequency between samples. See also the discussion on adaptive frequency sampling at the FR card.

If the imported `*.rsd` file contains currents for several frequencies, the option *New source* must be chosen as the AC card then results in a frequency loop and currents with different

frequencies cannot be superimposed. (If it is not chosen, PREFEKO will give an error). The frequency is defined in the `*.rsd` file, thus the preceding FR cards are ignored when processing an AC card.

All commands following the AC card in the FEKO input file (for example FF, FE, OS, GF, BO, ...) are processed within a frequency loop through all the frequencies in the `*.rsd` file. The loop is terminated by any of the following three cards (these cards are not included in the loop they terminate)

- AC (importing a new `*.rsd` file, or using the flag $I_2 = 0$)
- FR (manually setting a new frequency)
- EN (end of the FEKO input file)

For example, if a CableMod file must be read and the near field calculated for each frequency, the input file may look as follows

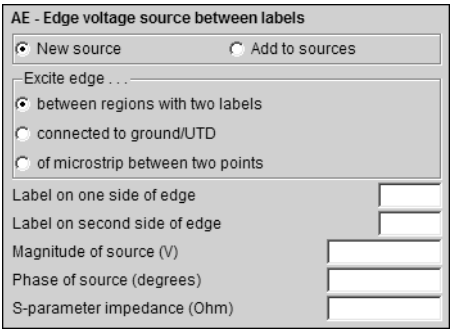
```
AC  ...      ** Read the *.rsd file
FE  ...      ** Calculate the near field
EN           ** End
```

However, if one wants to analyse, for example, a metal plate, which is excited first by an impressed line current and then also by a plane wave (in each case the near fields and the currents on the plate must be written to the output file), the input file would be

```
** Excitation by a line current
AC  ...      ** Read the *.rsd file
FE  ...      ** Calculate the near field
OS  ...      ** Output the currents
** Excitation by a plane wave
FR  ...      ** Set the frequency and terminate
AC loop
A0  ...      ** Specify plane wave excitation
FE  ...      ** Calculate the near field
OS  ...      ** Output the currents
** End EN
```

10.2.12 AE card

This card specifies an excitation at an edge between triangular surface elements similar to the A7 card, but the AE card has the advantage that the location of the feed point and the positive feed direction are substantially easier to specify. In addition it is possible to specify a feed edge which contains a number of triangle edges as shown in figure 10-4.


AE - Edge voltage source between labels
New source
Add to sources
Excite edge . . .
between regions with two labels
connected to ground/UTD
of microstrip between two points
Label on one side of edge
Label on second side of edge
Magnitude of source (V)
Phase of source (degrees)
S-parameter impedance (Ohm)

Parameters:

New source: New excitation, replaces all previous excitations.

Add to sources: Additional excitation, is added to previous excitations.

Excite edge: This specifies how the edge is determined:

- *between regions with two labels:* The excitation is placed on the edge between the regions with labels specified in *Label on one side of edge* and *Label on second side of edge*. The positive source direction is towards the *second* side.
- *connected to ground/UTD:* Excite the edges of metallic triangles with the label specified in the *Label of triangles connected to ground,* which are connected to UTD surfaces or to a PEC ground plane (as specified with a BO or GF card). The positive feed direction is towards the UTD region or ground plane.
- *of microstrip between two points:* This is a special microstrip line feed. The excitation is placed on all edges on a line between points (previously specified with DP cards). The points are specified in the *Start point of edge* and *End point of edge* dialogs. A GF card with a conducting ground must be present. The positive source direction is from the triangles towards the edge.

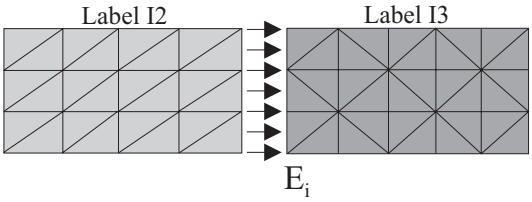


Figure 10-4: Example of the use of the AE card

Magnitude of source:	Magnitude value of the voltage $\|U_0\|$ in V.
Phase of source:	Phase of the voltage U_0 in degrees.
S-parameter impedance:	The reference impedance to use for this excitation if it is included in an S-parameter calculation requested by the SP card. If this field is empty or 0, the impedance specified at the SP card is used. (This value is only used if an SP card is applied to this source.)

The positive source direction as used above is in the direction of the current flow through the edge. The internal EMF (electromagnetic force) of the impressed voltage source is in the opposite direction.

It should be noted that the edge between the surfaces with labels *Label on one side of edge* and *Label on second side of edge* does not have to be straight. One may, for example, excite two half cylinders against each other. If an impedance must also be applied to the edge, the AE card can be combined with the LE card.

For the case where the item, *between regions with two labels,* is selected, more than two triangles may be connected to each edge section as shown in figure 10-5. However, one of these triangles must have a unique label and all the other triangles connected to this edge must have the same label. In figure 10-5 there is one triangle with label 2 at each edge between triangles and all the others have label 1. One may therefore specify an edge between labels 1 and 2. The edge in figure 10-6 cannot be used as a feed edge as there are more than one triangle of each of the two labels involved. The edge shown in figure 10-7 cannot be used as a feed edge either as there are triangles with three different labels connected to each edge.

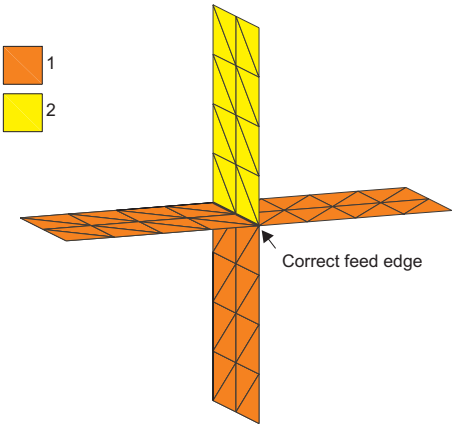


Figure 10-5: Example of a feed edge where more than two triangles are connected

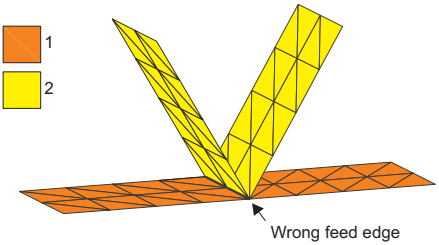


Figure 10-6: Disallowed feed edge (there is no triangle with a unique label at the edge)

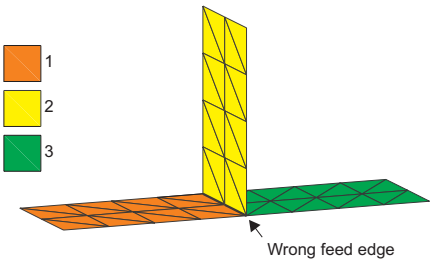
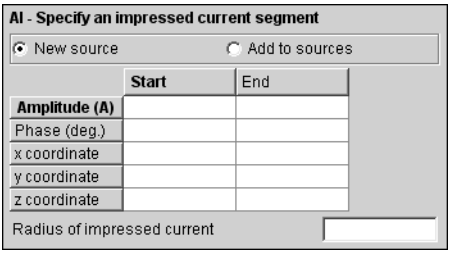


Figure 10-7: Disallowed feed edge (triangles with three different labels join at the edge)

10.2.13 AI card

With this *two line* card an impressed current source is specified. The current varies linearly between the values at the start and end points, see figure 10-8.



Parameters:

New source: New excitation, replaces all previous excitations.

Add to sources: Additional excitation, is added to previous excitations.

Amplitude: Amplitude $|I_1|$ in A of the current at the *Start* point, $\vec{r}_1$, and *End* point, $\vec{r}_2$.

Phase: Phase of the current at the start and end points in degrees.

x, y, z coordinate: Coordinates of the start and end points in m. (Note that all the coordinate values are optionally scaled by the SF card.)

Radius of impressed current: This parameter is optional. If specified, and different from zero, this value gives a finite wire radius for the impressed current element. FEKO then assumes that the current is uniformly distributed on the wire surface and uses the exact wire integral. If this parameter is not specified, the current filament approximation is used. (This value is optionally scaled by the SF card.)

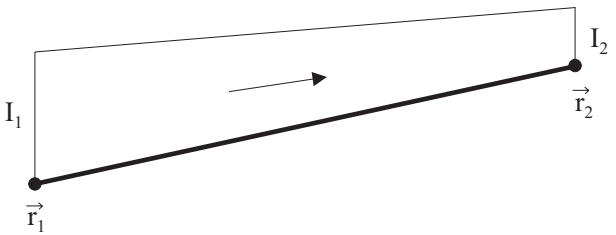


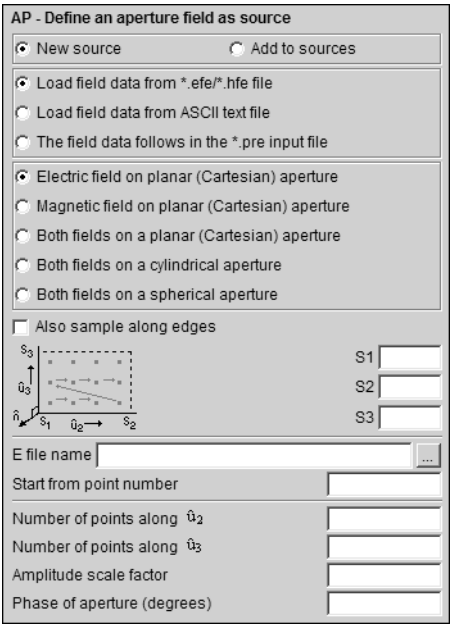
Figure 10-8: Impressed line current with a linear current distribution

The following restrictions apply when using the impressed current elements.

- It is not possible to attach the impressed current to a wire segment in the FEKO model. (If the impressed current is making electrical contact with a triangular surface current element, the AV card should be used.)
- When modelling dielectric bodies with the surface equivalence method, the current element must be in the free space medium, i.e. outside the dielectric bodies. (The material parameters of this medium can, however, be set with the EG and/or GF cards).
- When used with the spherical Green's function, the current element must be outside the dielectric spheres.
- The current segments may be joined with each other and with the AV card to form long paths and/or closed loops. The point charges which arise when the current does not go to zero at an end point or when there is a current discontinuity at a connection point, are not taken into consideration. This is required to model, for example, the case where radiating lines are terminated in a non-radiating structure. If these charges must be considered explicitly, the line current should be modelled by a row of Hertzian dipoles (see the A5 card). Note, however, that the constant line charge along the current segment is correctly taken into account.
- If several of these current elements are used, the total radiated power (required to calculate, for example, the far field gain/directivity) can only be calculated accurately if the mutual coupling between segments is taken into account. Due to neglecting the point charges at the ends of the segments, the coupling cannot be determined accurately. If exact values of the radiated power are required, it should be determined by integrating the far field (see the FF card). It should be noted that, for example, the computed near and far fields (the actual field strength values), the induced currents, coupling factors and losses are computed correctly.

10.2.14 AP card

With this card a planar, cylindrical or spherical aperture of measured or calculated field values is converted into an equivalent array of electric and magnetic dipoles. The card is processed by PREFEKO and replaced by A5 and A6 cards in the `*.fek` file.


AP - Define an aperture field as source
New source
Add to sources
Load field data from *.efe/*.hfe file
Load field data from ASCII text file
The field data follows in the *.pre input file
Electric field on planar (Cartesian) aperture
Magnetic field on planar (Cartesian) aperture
Both fields on a planar (Cartesian) aperture
Both fields on a cylindrical aperture
Both fields on a spherical aperture
Also sample along edges
S1
S2
S3
E file name
Start from point number
Number of points along û2
Number of points along û3
Amplitude scale factor
Phase of aperture (degrees)

Parameters:

New source: New excitation, replaces all previous excitations.

Add to sources: Additional excitation, is added to previous excitations.

. . . field data . . . : In this group one selects between *Load field data from *.efe/*.hfe file* (i.e. from `*.efe` or `*.hfe` files previously calculated with FEKO), *Load field data from ASCII text file* (the format of these files are documented below) or *The field data follows in the *.pre input file* (described below).

. . . aperture: Here one may select a planar, cylindrical or spherical aperture. For a planar aperture one may elect to use only electric or magnetic fields (this radiates in both directions, see comment below).

Also sample along edges: This item is used to determine if dipoles should lie on the edges of the aperture or not (see figures 10-9 to 10-12). When checked the outer dipoles lie on the edges, when unchecked the dipoles lie half an increment away from the edges. Dipoles should not lie on

the edges of two apertures that have a common side, otherwise two dipoles may have the same location and polarisation. If this is the case the power calculation in FEKO will fail.

S1, S2 and S3: These define the orientation of the aperture and should be clear from the figure on the dialog. For a planar aperture it defines the position of the origin and the direction of the $\hat{u}_2$ and $\hat{u}_3$ directions respectively. (The field data is assumed to vary first along the $\hat{u}_2$ direction.) For cylindrical and spherical apertures they define the origin and the direction of the local z and x axes respectively. All angles are relative to these axes (see figures 10-11 and 10-12).

E file name: The input filename (`*.efe` or text file), from which the field data must be read.

H file name: The input filename (`*.hfe` or text file), from which the field data must be read.

Start from point number: The number of the first field point to be used for the aperture. If set to 1, field values are read from the start of the file, for larger values the first point number-1 values (`*.efe` and `*.hfe` files) or lines (text files) are ignored. This may be used, for example, if the data file contains the field values for more than one aperture. The *Start from point number* field is not used if the field data is obtained from the `*.pre` input file.

Number of points along . . . : These two fields specify the number of field points along each of the two axes of the aperture.

Amplitude scale factor: A constant by which the amplitudes of all the dipoles in the aperture are scaled.

Phase of aperture: constant phase added to all dipoles in the aperture.

The aperture is based on the equivalence principle. This states that the sources and scatterers inside a given volume can be removed, and modelled by placing the equivalent currents $\vec{J}_s = \hat{n} \times \vec{H}$ and $\vec{M}_s = -\hat{n} \times \vec{E}$ on the enclosing surface. The vector $\hat{n}$ is a unit vector, normal to the surface, and points towards the exterior region. The fields in this region are the same as the original fields, while those in interior region are zero.

Field values are read from the data files (with a possible offset specified with *Start from point number*) or the `*.pre` input file and converted to equivalent electric (magnetic fields) and magnetic (electric field) dipoles at these points. *Note that all angles are read from the data, but no distance values.* Thus for planar apertures the positions are calculated entirely from the specified points (S1, S2 and S3). For cylindrical apertures S1 and S2 specify the extents of the aperture along the local $\hat{z}$ direction and S1–S3 specifies the direction of the x axis as well as the radius of the cylinder. The points are placed at the φ-values listed together with the field data. For spherical apertures, S1–S2 specifies the direction of the z axis and S1–S3 the x axis. S2 and S3 must lie on the same radius which is also the radius of the field points. In this case both ϑ and φ are read with the data.

Figures 10-9 and 10-10 show the application of the equivalence principle to a planar aperture. In both figures there are the same number of field points along the two orthogonal directions. When the *Also sample along edges* item is checked, the first point lies at S1 with the following points in the direction of S2 as shown by the indices in figure 10-9. When it is unselected, the pattern is as shown in figure 10-10. The normal vector is calculated from $\hat{n} = \hat{u}_2 \times \hat{u}_3$ with $\hat{u}_2$ and $\hat{u}_3$ as defined in the figure.

Figure 10-11 shows the dipole locations for a cylindrical aperture created from a data file containing field values for φ from 20° to 80° in 10° increments and 5 values in the z direction. When the *Also sample along edges* item is checked, the samples extend up to the edges of the aperture, the points and the effective aperture are shown in figure (a). When it is unchecked, samples do not lie on the edges as shown in figure (b). Note that, when using identical input data as for the case when the item is checked, the z positions of the samples changed, while in the φ direction the size of the effective aperture is increased by 5° on both sides.

Figure 10-12 shows the dipole locations for a spherical aperture created from field values

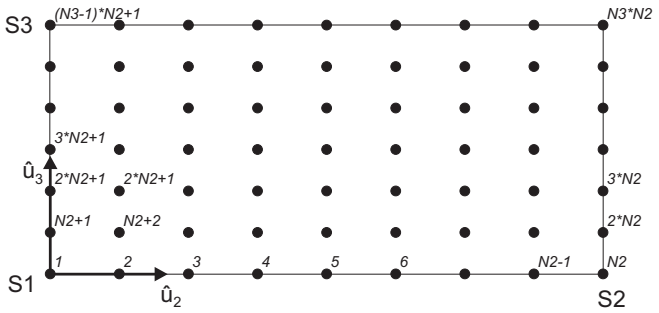


Figure 10-9: Location of the equivalent dipoles on a planar aperture where the *Also sample along edges* item is checked.

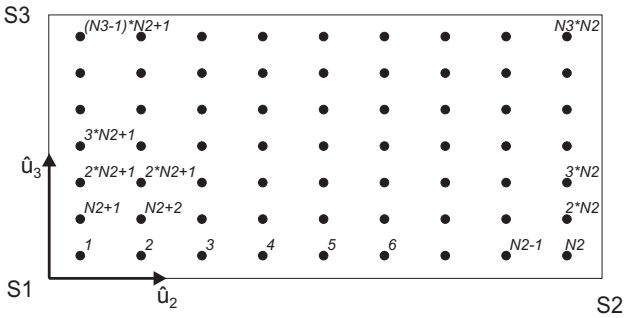


Figure 10-10: Location of the equivalent dipoles on a planar aperture where the *Also sample along edges* item is unchecked.

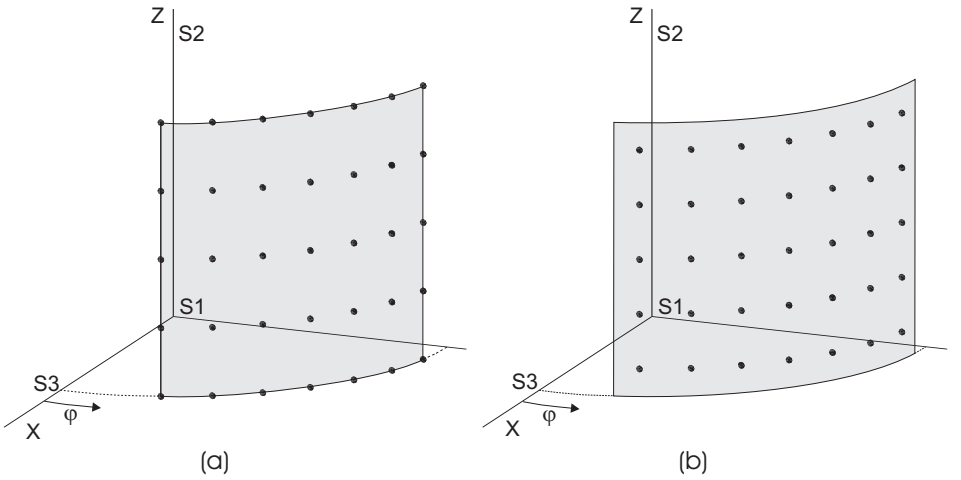


Figure 10-11: Location of the equivalent dipoles on a cylindrical aperture.
(a) *Also sample along edges* checked (b) *Also sample along edges* unchecked.

for ϑ from 40° to 80° with 10° increments and φ from 20° to 80° also with 10° increments. In this case the aperture increases in size in both directions when the *Also sample along edges* item is checked.

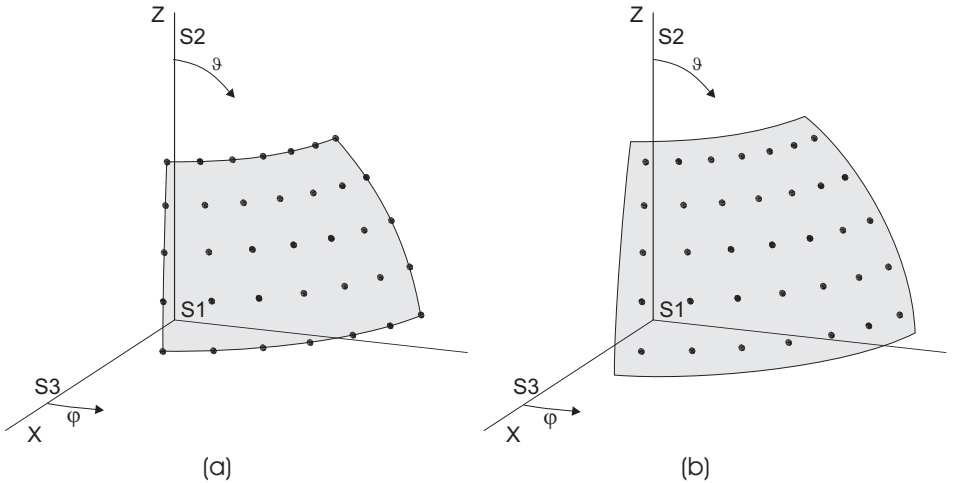


Figure 10-12: Location of the equivalent dipoles on a spherical aperture.
(a) *Also sample along edges* checked (b) *Also sample along edges* unchecked.

For planar apertures, the data must vary first along the $\hat{u}_2$ direction. For cylindrical and

spherical apertures PREFEKO will determine which coordinate is incremented first and write out the dipoles accordingly.

The dipole amplitude is the product of the surface current and the incremental area between samples. In addition, the amplitude of the dipoles on the sides (when the *Also sample along edges* item is checked) are reduced by a factor of 2 and those on the corners by a factor of 4 such that the effective aperture of integration has the same size as the specified aperture.

A fully closed surface can be created by specifying 6 planar apertures or a spherical one. The surface equivalence principle can be applied to this surface by reading both electric and magnetic fields for each plane. (For planar apertures the user should specify 6 AP cards, each using both electric and magnetic fields. If separate cards are used for the electric and magnetic fields the radiated power is not calculated correctly.) The normal vector must point to the exterior region, normally this is outward. (For planar apertures created form `*.efe` and `*.hfe` files, the sample order determines the directions of $\hat{u}_2$ and $\hat{u}_3$ which, in turn, determines the normal vector $\hat{n} = \hat{u}_2 \times \hat{u}_3$. If this is pointing into the cube, an additional 180° phase shift is obtained by setting *Phase of aperture (degrees)* to 180. This changes the sign of the field radiated by the aperture which, when interacting with the remaining sources, will result in the correct total fields in the desired region.) All surfaces and scatterers inside the body must be removed and those outside retained.

For planar apertures (for example, the opening of a horn antenna), one may use the mirror principle if the field at the edges can be neglected. This results in a duplication of the magnetic current and cancellation of the electric current. Thus it is sufficient to read only the electric fields and scale by the factor *Amplitude scale factor=2*. In this case any sources or structures in the region towards which the normal is pointing, should also be subjected to the mirroring (i.e. the structures should be electrically mirrored by using the SY card). Further, it should be remembered that the fields will only be correct in the direction that the normal vector points to. The symmetric fields in the other half-space will not be equal to the fields of the original problem. Note that FEKO takes this into account and divides the total radiated power by two when calculating the power radiated by a planar aperture containing only electric or magnetic fields.

When the data is read from an ASCII format text file, each line in the file represents one point and the values are space delimited. For planar apertures it must have four parameters: The absolute value and phase (in degrees) of the field component in the $\hat{u}_2$ direction followed by the absolute value and phase in the $\hat{u}_3$ direction (see the example below). The data must be such that the position increments along the $\hat{u}_2$ direction first. For cylindrical apertures it must have five parameters: The angle φ (in degrees) followed by the absolute value and phase of the $\hat{\varphi}$ component and the absolute value and phase of the $\hat{z}$ component. For spherical apertures it must have six parameters: The angles ϑ and φ followed by the absolute value and phase of the $\hat{\vartheta}$ and $\hat{\varphi}$ components.

When the data is read from the `*.pre` input file, the data must be in the normal column based input format and FOR loops etc. may be used. The four field components are the same as for the text data, and must be entered in columns 10 characters wide from columns 51 to 90. The angle φ must be in columns 40 to 49 and ϑ in columns 30 to 39 when they are required. If both electric and magnetic fields are required, all the electric fields are given first followed by the same number of magnetic fields.

Example of AP card usage:

As an example consider an open ended X-band waveguide radiating through a hole in a large ground plane as shown in figure 10-13. Away from the aperture the plane $z = 0$ is perfectly conducting, i.e. the tangential electric field is zero, while the magnetic field is not — thus we will use electric symmetry.

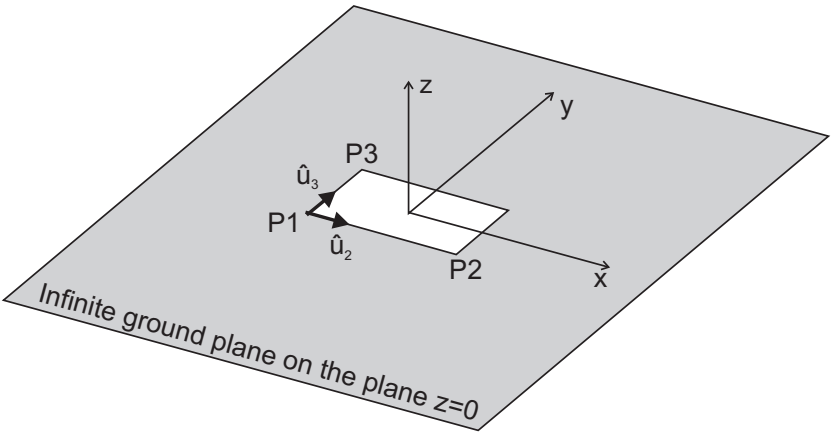


Figure 10-13: Example of an open waveguide as an implementation of the AP card

For this example the field is considered to be purely y directed (i.e. it has only a $\hat{y}$, or $\hat{u}_3$, component). The field is assumed to be constant in the y direction and to have a cosine distribution in the x direction (i.e. the $\hat{u}_2$ axis).

With *Number of points along* $\hat{u}_3$=5 and *Number of points along* $\hat{u}_2$=3 — in practice more points may be required — the data file will be as follows:

```
0.0     0.0     0.0     0.0
0.0     0.0     0.707   0.0
0.0     0.0     1.0     0.0
0.0     0.0     0.707   0.0
0.0     0.0     0.0     0.0
0.0     0.0     0.0     0.0
0.0     0.0     0.707   0.0
0.0     0.0     1.0     0.0
```

```
   0.0    0.0    0.707  0.0
   0.0    0.0    0.0    0.0
   0.0    0.0    0.0    0.0
   0.0    0.0    0.707  0.0
   0.0    0.0    1.0    0.0
   0.0    0.0    0.707  0.0
   0.0    0.0    0.0    0.0
```

The zero values will not result in any dipoles, but they must be in the data file to allow correct indexing. The `*.pre` file will contain the following section:

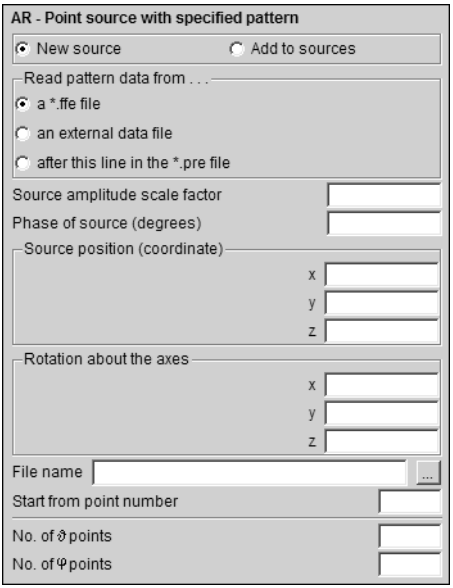
```
...

** Only electric fields --- use electric symmetry in the z=0 plane
SY   1    0    0    2

** Define the corner points of the aperture
#wx = 0.02286
#wy = 0.01016
DP     P1                     -#wx/2    -#wy/2     0
DP     P2                      #wx/2    -#wy/2     0
DP     P3                     -#wx/2     #wy/2     0

** The geometry ends after the corner nodes have been defined
EG   1    0    0    0    0

** Specify the frequency
FR   1    0                   9.375e9

** Specify the AP card as a new source
** The amplitude factor of 2.0 is due to the use of the equivalence principle
AP   0    3    P1   P2   P3   1        5        3        2.0      0.0    "Guide.dat"

...
```

which will generate nine x directed magnetic dipoles of the correct magnitude in the `*.fek` file.

10.2.15 AR card

With this card the radiation pattern of an antenna is used as an impressed source. The pattern is read from a data file or defined in the `*.pre` file below the AR card.

This card has a variety of uses, for example, importing measured radiation patterns, synthesising arrays from the individual patterns of the elements, realising radiation only within certain sectors, etc. In the MoM/UTD hybrid it is possible to simulate, for example, the antenna on its own and to save the far field in a `*.ffe` file. This field is then imported and used as source in the UTD part which may greatly speed up the ray tracing computation as there is now only one source point.



Parameters:

New source: New excitation, replaces all previous excitations.

Add to sources: Additional excitation, is added to previous excitations.

Read pattern data from: In this group the user must select one of three options:

- *a *.ffe file:* Read the radiation pattern from an `*.ffe` format file (which may be created with the DA and FF cards).
- *an external data file:* Read the radiation pattern from an ASCII file (the format of this file is described below).
- *after this line in the *.pre file:* The radiation pattern is specified in the `*.pre` file following the AR card (the format is described below). With this option one can make use of the FOR loops to generate patterns from known functions.

Source amplitude scale factor: The field strength values in each direction is determined from the data. This parameter is used to scale the amplitude of the entire pattern by a constant value.

Phase of source: This parameter specifies a constant additional phase for the entire pattern.

Source position: In this group the x, y and z coordinates of the source point (i.e. the position where the antenna is placed) is entered in m. This value is affected by the scale factor of the SF card if used.

Rotation about the axes: In this group the angle with which the imported pattern is rotated around the x, y and z axis is entered. We will refer to these fields as α_x, α_y and α_z in the rest of this discussion.

Filename: The name of the `*.ffe` or ASCII input file.

Start from point number: This parameter is only relevant when the data is read from an external file, and gives the line number of the first line to read from the input file. If the data must be read from the beginning of the file, the value in this field should be set equal to 1. This parameter is used when the `*.ffe` file contains more than one pattern. For example, if the file contains the pattern at various frequencies, the correct pattern can be selected by setting this field to the appropriate value for each frequency.

Number of ϑ points: The number of ϑ angles in the pattern.

Number of φ points: The number of φ angles in the pattern.

The radiation pattern of the antenna must be specified in spherical coordinates (ϑ, φ) with the phase centre located at the origin of the local coordinates (as used in the pattern data). If this is not the case, the phase of the far fields will not be correct. (For example, if a `*.ffe` file is exported with FEKO to be used with the AR card, the origin should be shifted with the OF card to the phase centre of the antenna.) The vertical and horizontal components of the complex electric field E_ϑ^{FF} and/or E_φ^{FF} must be specified at discrete angles (ϑ_i, φ_j) with i and j larger or equal to 1 and smaller or equal to the number of points specified for the respective angles. The field values are entered in Volts and the actual far fields are calculated from

$$E_\vartheta = E_\vartheta^{FF} \cdot \frac{e^{-j\beta R}}{R} \quad \text{or} \quad E_\varphi = E_\varphi^{FF} \cdot \frac{e^{-j\beta R}}{R}$$

with R the distance to the field point and β the complex propagation constant in the free space medium (see the EG and GF cards). These formulas are used for all distances R, i.e. also in the near field. However FEKO tests whether the far field conditions are met (by calculating the directivity and equivalent aperture) and gives an appropriate warning if this is not the case.

The permissible range of the angles ϑ_i is $0° \ldots 180°$ and they must be in ascending order, i.e. $\vartheta_{i+1} > \vartheta_i$. However, the angles do not have to be equidistantly spaced. (Thus, for example, for a highly directive antenna, a denser grid can be used close to the main beam direction.) The same applies to the angles φ_j where the permissible range is $0° \ldots 360°$. For field angles outside the start and end values defined in the data (i.e. for $\vartheta < \vartheta_1$, $\vartheta > \vartheta_{\max}$, $\varphi < \varphi_1$ or $\varphi > \varphi_{\max}$), the field strengths E_ϑ^{FF} and E_φ^{FF} are set to zero, such that a sector radiator can be realised. The values at field angles within the defined range are determined by bilinear interpolation. To realise a complete radiation pattern, rather than a sector radiator, the angles should be defined such that $\vartheta_1 = 0°$, $\vartheta_{\max} = 180°$, $\varphi_1 = 0°$ and $\varphi_{\max} = 360°$.

The radiation pattern, specified in the local spherical coordinate system (ϑ, φ) of the antenna, is read and initially placed at the origin of the global coordinate system in which the `*.pre` file is constructed. The pattern is now rotated by an angle α_z around the z axis, by α_y around the y axis and by α_x around the x axis. (The rotation is identical to the rotation executed by the TG card in section 9.2.38 and the rotation matrix M is applicable to both the TG and AR cards.) Finally the pattern is shifted to the specified location.

If the AR card is used simultaneously with a ground plane (BO card), FEKO includes the influence of the ground plane on the radiation pattern. The imported pattern must therefore be the free space radiation pattern of the antenna (in the absence of the ground plane). If this is not the case the influence of the ground plane is considered twice.

The use of the PW card to specify the radiated power is allowed. The field amplitudes $|E_\vartheta^{FF}|$ and $|E_\varphi^{FF}|$ will be scaled accordingly. Multiple radiation patterns can be used simultaneously, and also with other sources such as an incident plane wave. In such a case, the coupling is not considered when the radiated power is determined.

The AR card cannot be used with special Green's functions for a layered sphere or for a layered substrate.

The format of the data depends on the value of the parameter *Read pattern data from:*

- *an *.ffe file* (`*.ffe` file)

  With this option, the radiation pattern is read from a `*.ffe` file created with FEKO (using the DA and FF cards). All the data of the radiation pattern (angles and field values) are determined from the file. The user should ensure that, for example, the frequency is correct. If an antenna is analysed with FEKO, the far field can be exported to the `*.ffe` file using the commands (for 5° angle increments)

```
DA   0    0    1    0    0
FF   1    37   73   0         0         0         5         5
```

Note 37 points are used for ϑ and 73 for φ to ensure that the radiation pattern is closed (see also the comment above).

This can then be imported as a source into another model with the command

```
AR   0    1    1    37   73   1.0       0.0       0.0       0.0       0.0 ...
                                ...   0.0       0.0       0.0     "file.ffe"
```

- *an external ASCII file*

  With this option, the data is read from the specified external data file. Each line contains 6 space delimited data fields in the following order:

  - The angle ϑ in degrees
  - The angle φ in degrees
  - Amplitude of the field strength E_{ϑ}^{FF} in V
  - Phase of the field E_{ϑ}^{FF} in degrees
  - Amplitude of the field strength E_{φ}^{FF} in V
  - Phase of the field E_{φ}^{FF} in degrees

  The inner loop should be with respect to the angle ϑ such that the order of the lines is as follows

$$\begin{array}{ccc} \vartheta_1 & \varphi_1 & \cdots \\ \vartheta_2 & \varphi_1 & \cdots \\ \vartheta_3 & \varphi_1 & \cdots \\ \vdots & \vdots & \cdots \\ \vartheta_{I_4} & \varphi_1 & \cdots \\ \vartheta_1 & \varphi_2 & \cdots \\ \vartheta_2 & \varphi_2 & \cdots \\ \vdots & \vdots & \cdots \\ \vartheta_{I_4} & \varphi_{I_5} & \cdots \end{array}$$

- *after this line in the *.pre file*

  This case is similar to reading an external ASCII file, except that the data is read directly from the `*.pre` input file. The six data fields mentioned for the case of an ASCII file must appear in the 6 columns of 10 characters starting at character 31 and ending at character 90 in the lines following the AR card. When this option is selected, the card dialog shows additional input fields where the user can specify these values for each point. The data lines may be separated by comment lines (EDITFEKO, however, does not support this) and FOR–NEXT loops may be used. Even when using FOR loops one may use the card dialog in EDITFEKO to generate a typical line.

We conclude this description with an example of a sector radiator. We want to realise an ideal sector radiator, which radiates 10 Watt horizontal polarisation in the angular region defined by $-70° \leq \varphi \leq 70°$ and $75° \leq \vartheta \leq 105°$. Since the angle range of the imported pattern must be positive one may define separate sources for the regions $0° \leq \varphi \leq 70°$ and $290° \leq \varphi \leq 360°$. A more elegant solution is to define a single pattern in the range $0° \leq \varphi \leq 140°$ and rotate it by $-70°$ around the z axis. The complete radiation pattern is defined in the following input file (note that only horizontal polarisation, i.e. E_φ^{FF}, is required):

```
** Application example for the AR card: Sector radiator

** No other structures considered
EG   1   0   0   0   0

** Set the frequency
FR   1   0               100.0e6

** Specified radiated power
PW   1                   10.0

** Define the sector radiator
AR   0   3       2   2   1.0      0.0      0.0      0.0      0.0      0.0      0.0      -70
**                       Theta    Phi       E_theta           E_Phi
                         75       0        0        0        1        0
                         105      0        0        0        1        0
                         75       140      0        0        1        0
                         105      140      0        0        1        0
** Check: Compute the full 3D radiation pattern with 5 deg stepping
FF   1   37  73  0        0.0      0.0      5.0      5.0

** End
EN
```

FEKO determines a directivity of 10.1 dBi. The radiation pattern is easily validated by calculating the far field as shown with the FF card in the last step. The result is shown in figure 10-14.

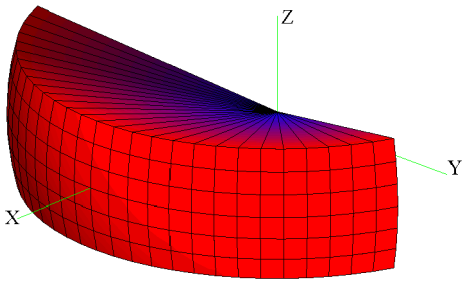
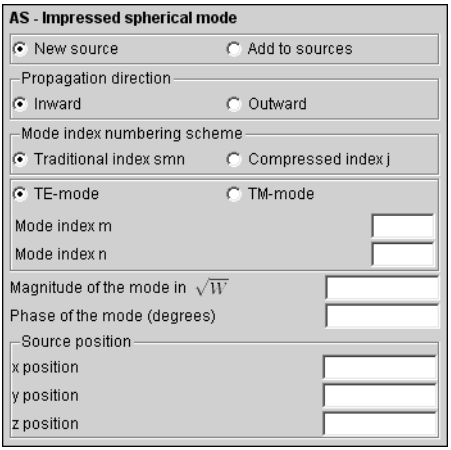


Figure 10-14: 3D radiation pattern of the sector radiator.

10.2.16 AS card

This card specifies an excitation by means of impressed spherical modes, which are either radiating (i.e. propagating in positive r direction to infinity, with r being the radius in a spherical coordinate system) or incident onto a structure (i.e. then propagating towards the origin $r = 0$).



This excitation option can thus be used for both the synthesis of an arbitrary electromagnetic field (sum of the modes weighted with complex mode coefficients), and also for the determination of the response (i.e. induced voltage or power at a load) of a receiving antenna due to the incident modes (leading to the so-called generalised scattering matrix).

Parameters:

New source: New excitation, replaces all previous excitations.

Add to sources: Additional excitation, is added to previous excitations.

Propagation direction: This determines if the spherical waves propagate *Inward* (the model is illuminated with modes propagating towards $r = 0$, i.e. spherical Hankel function of the first kind $z_n^{(3)=}h_n^{(1)}$) or *Outward* (the modes radiate towards $r = \infty$, i.e. spherical Hankel function of the second kind $z_n^{(4)} = h_n^{(2)}$).

Traditional index smn: If this option is checked, the user can specify *TE-mode* ($s = 1$) or *TM-mode* ($s = 2$) and the indices m and n in the group below. Here n is the mode index in radial direction and must be in the range $1, 2, \ldots \infty$ and m is the mode index in the azimuth direction φ. We do not distinguish between *even* and *odd* modes (with $\cos(m\varphi)$ and $\sin(m\varphi)$ angular dependencies), but rather use the angular dependency $e^{jm\varphi}$. Thus the index m can also be negative, but it must be in the range $-n \ldots n$.

Compressed index j: With this option, a compressed one-dimensional mode numbering scheme is used. The *Mode index j* is then specified in the field below. Here

$$j = 2\,[n\,(n+1) + m - 1] + s$$

where $s = 1$ for TE-modes and $s = 2$ for TM-modes. This unified mode numbering scheme allows the computation of an extended scattering matrix (with network and radiation ports). This index j then represents a unique port number in the scattering matrix.

Magnitude of the mode in $\sqrt{W}$: Absolute value of the complex amplitude of this specific spherical mode (due to the applied normalisation of the spherical modes, the unit of this amplitude is $\sqrt{W} = \sqrt{VA}$).

Phase of the mode: The phase of the complex amplitude of this spherical mode in degrees.

Source position: The position of the origin $r = 0$ of the mode in m. (These values are optionally scaled by the SF card.)

The implementation of the spherical modes at the AS card follows closely the references: J. E. Hansen, *Spherical Near–Field Antenna Measurements*, Peter Peregrinus Ltd., London, 1988 and B. C. Brock, *Using Vector Spherical Harmonics to Compute Antenna Mutual Impedance from Measured or Computed Fields*, Sandia National Laboratories, Report SAND2000-2217-Revised, April 2001. One must realise that Hansen assumes a complex time dependence of $e^{-i\omega\,t}$, while FEKO always uses the positive sign $e^{j\omega\,t}$.

In FEKO, using the modal coefficients Q^c_{smn} the electric and magnetic field strength is represented in a spherical coordinate system by

$$\vec{E}(r,\varphi,\vartheta) = \sum_{c=3}^{4}\sum_{s=1}^{2}\sum_{n=1}^{\infty}\sum_{m=-n}^{n} Q^{(c)}_{smn}\vec{F}^{(c)}_{smn}(r,\varphi,\vartheta)$$

$$\vec{H}(r,\varphi,\vartheta) = \tfrac{j}{Z_F}\sum_{c=3}^{4}\sum_{s=1}^{2}\sum_{n=1}^{\infty}\sum_{m=-n}^{n} Q^{(c)}_{smn}\vec{F}^{(c)}_{3-s,m,n}(r,\varphi,\vartheta)\ .$$

Here s, m and n are the mode indices with $s = 1$ indicating the TE-mode and $s = 2$ the TM-mode (see also the discussion above), and c represents the propagation direction: $c = 3$ is inward and $c = 4$ is outward. The term Z_F denotes the wave impedance of the medium under consideration, β below is the corresponding wavenumber.

The spherical wave functions $F^{(c)}_{smn}$ are given by

$$\begin{aligned}\vec{F}^{(c)}_{1mn}(r,\vartheta,\varphi) = \vec{M}^{(c)}_{mn} &= \beta\sqrt{\tfrac{Z_F}{2\pi}}\,\tfrac{1}{\sqrt{n(n+1)}}\left(-\tfrac{m}{|m|}\right)^m \\ &\cdot\Big(0\cdot\vec{e_r} \\ &+z^{(c)}_n(\beta r)\,\tfrac{jm}{\sin\vartheta}\,\widehat{P^{|m|}_n}(\cos\vartheta)\,e^{jm\varphi}\,\vec{e_\vartheta} \\ &-z^{(c)}_n(\beta r)\,\tfrac{\partial}{\partial\vartheta}\left\{\widehat{P^{|m|}_n}(\cos\vartheta)\right\}e^{jm\varphi}\,\vec{e_\varphi}\Big)\end{aligned}$$

and

$$\begin{aligned}\vec{F}^{(c)}_{2mn}(r,\vartheta,\varphi) = \vec{N}^{(c)}_{mn} &= \beta\sqrt{\tfrac{Z_F}{2\pi}}\,\tfrac{1}{\sqrt{n(n+1)}}\left(-\tfrac{m}{|m|}\right)^m \\ &\cdot\Bigg(\tfrac{n(n+1)}{\beta r}\,z_n^{(c)}(\beta r)\,\widehat{P_n^{|m|}}(\cos\vartheta)\,e^{jm\varphi}\,\vec{e_r} \\ &+\tfrac{1}{\beta r}\,\tfrac{\partial}{\partial(\beta r)}\left\{\beta r\,z_n^{(c)}(\beta r)\right\}\,\tfrac{\partial}{\partial\vartheta}\left\{\widehat{P_n^{|m|}}(\cos\vartheta)\right\}\,e^{jm\varphi}\,\vec{e_\vartheta} \\ &+\tfrac{1}{\beta r}\,\tfrac{\partial}{\partial(\beta r)}\left\{\beta r\,z_n^{(c)}(\beta r)\right\}\,\tfrac{jm}{\sin\vartheta}\,\widehat{P_n^{|m|}}(\cos\vartheta)\,e^{jm\varphi}\,\vec{e_\varphi}\Bigg)\end{aligned}$$

with the associated Legendre function

$$\widehat{P_n^{|m|}}(\cos\vartheta) = \sqrt{\frac{2n+1}{2}\frac{(n-|m|)!}{(n+|m|)!}}\,P_n^{|m|}(\cos\vartheta)$$

and the spherical Bessel functions

$$\begin{aligned}z_n^{(3)}(\beta r) &= h_n^{(1)}(\beta r) = j_n(\beta r) + j\,y_n(\beta r) \\ z_n^{(4)}(\beta r) &= h_n^{(2)}(\beta r) = j_n(\beta r) - j\,y_n(\beta r)\ .\end{aligned}$$

It should be noted that the Legendre polynomial $P_n^{|m|}(\cos\vartheta)$ as used in FEKO follows the definitions of Abramowitz/Stegun (also used like this in Numerical Recipes) or also Harrington. The formulas used in other references (e.g. Stratton or Hansen) have an extra factor $(-1)^m$ included. This is not considered in FEKO, and thus the mode coefficients Q^c_{smn} might differ from those computed according to Hansen (there is also of course the other time dependency, see earlier discussion).

Theoretically the index n runs in the range $1, 2, \ldots, \infty$. For any practical application, one will have to consider a finite number of modes only, i.e. limit the range $n = 1 \ldots N$. Some rules of thumb exist for the selection of N. For instance when representing the pattern of an antenna by spherical modes one can use the upper limit

$$N \approx \beta r_0 = 2\pi\frac{r_0}{\lambda}\ ,$$

where β is the wavenumber, λ the wavelength, and r_0 denotes the radius of the smallest sphere enclosing the antenna. In critical cases, one might also rather use

$$N \approx \beta r_0 + 10$$

or

$$N \approx \beta r_0 + 3\,\sqrt[3]{\beta r_0}\ .$$

When using the compressed numbering scheme with one index j, any upper limit N for n will with the largest values $m = N$ and $s = 2$ translate into an upper limit

$$J = 2\,[N\,(N+1) + N - 1] + 2 = 2\,N\,(N+2)$$

for j (i.e. $j = 1 \ldots J$ then). So for instance for an antenna with enclosing radius $r_0 = \frac{\lambda}{4}$ (then $\beta r_0 = 1.57$) when using the last of the three rules of thumb above, one would need roughly $N \approx 5$ or $J \approx 70$ modes, respectively. For $r_0 = \lambda$ these limits become already $N \approx 12$ and $J \approx 336$, and for $r_0 = 5\lambda$ one has to use $N \approx 41$ and $J \approx 3526$ modes.

The origin $r = 0$ of the modes can be set arbitrarily in space by means of the *Source position* field. However, no provision has been made to allow a rotation, i.e. in spherical coordinates the distance r of some point is measured with respect to the mode origin, but the angles ϑ and φ are always defined with respect to the global coordinate system. The reason why a rotation is not supported here for the AS card (as opposed to the A5 or A6 elementary dipoles) is that the modes inherently contain this radiation in an arbitrary direction. For instance, the TM–mode $n = 1, m = 0$ represents a z–oriented electric dipole (i.e. vertical), while the TM–modes $n = 1, m = \pm 1$ represent horizontally aligned dipoles. An arbitrary pattern orientation can be obtained by a suitable superposition of different modes.

The modes have been normalised such that each mode has a constant power flow through any spherical surface (either inwards or outwards). In principle one could use the PW card for this, but then power normalisation works only if there is not more than one mode active at the same time (when using the PW card, just the total radiated power of all the modes is determined, and then each mode is scaled with the same factor, so that the total radiated power is correct, but here we enforce a specified power for each individual mode). The power for each mode is independent of the mode indices

$$P = 0.5\,|\textit{Magnitude of the mode}|^2$$

(unit is correctly Watt since the amplitude has a unit $\sqrt{W}$). Since the modes are orthogonal, if multiple AS cards are active at the same time, the powers of the individual modes can just be added. Any other power corrections (such as due to metallic elements being in the vicinity) are not taken into account in FEKO.

If an AS excitation is used in connection with multiple different media, it should be noted that we assume outward propagating modes (when *Outward* is selected under *Propagation direction*) to originate from the *Source position,* i.e. the source is located in the medium where *Source position* is located, and its contribution will be zero in all other media. For inward propagating modes (when *Inward* is selected under *Propagation direction*) we assume them to originate at infinity, in the free-space medium 0, and such modes only contribute to this medium with index 0. In connection with the UTD, only outward propagating modes are allowed (they have a well defined source point), while inward propagating modes are not supported (neither a source point nor an incidence direction can be assigned to such modes).

When computing the far field with the FF card, then outward propagating modes are included normally, which is important when synthesising antenna patterns by means of spherical modes. However, for inward propagating modes the far field limit for $R \to \infty$

of the field strength with the factor

$$\frac{e^{-j\beta R}}{R}$$

split off does not exist (similar to the non-existent far field for an incident plane wave). Thus such inward propagating modes are excluded from any far field computation. This is not a problem, since normally inward propagating modes are applied when computing the generalised antenna scattering matrix, i.e. the response of a receiving antenna to some incident mode. Then one looks at these quantities:

- Induced voltage or power at the antenna terminals for the network ports (i.e. no far-field computation).
- Field scattered back, and decomposition of this field into spherical modes (far field ports). Here one needs the far field computation, but similar to an RCS computation with an incident plane wave, only the scattered far field is of interest, which can be obtained from the FF card without problems.

Note that all the modes (inwards and outwards propagating) are correctly included when doing a near field computation with the FE card (also for very large distances).

As an application example, we consider the TE-mode $n = 5$ and $m = 0$ and compute the far field pattern:

```
** Create the far-field radiation pattern of a spherical mode

** No geometry
EG   1    0    0    0    0

** Set the frequency
FR   1                        100e6

** Spherical mode indices
#s = 1      ** 1 = TE-mode, 2 = TM-mode
#n = 5      ** mode order n (with n=1, 2, 3, ...)
#m = 0      ** mode order m (with m=-n ... n)

**   Excitation
AS   0    4    #s   #m   #n   1         0

**   Compute the full far-field pattern
FF   1    91   37   0         0         0         2         10

** End
EN
```

The resulting pattern is shown in figure 10-15. From the FEKO output file one can see the correct radiated power of 0.5 Watt as obtained from the far field integration:

```
Integration of the normal component of the Poynting vector in the angular
grid DTHETA =    2.00 deg. and DPHI =   10.00 deg. (    3367 sample points)
  angular range THETA         angular range PHI        radiated power
 -1.00 .. 181.00 deg.       -5.00 .. 365.00 deg.      5.13889E-01 Watt
  0.00 .. 180.00 deg.        0.00 .. 360.00 deg.      5.00001E-01 Watt
```

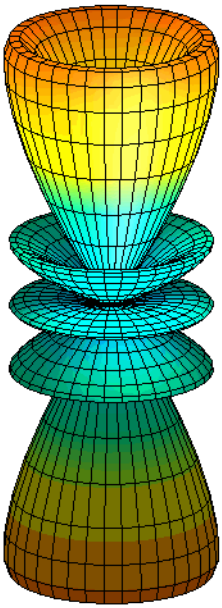
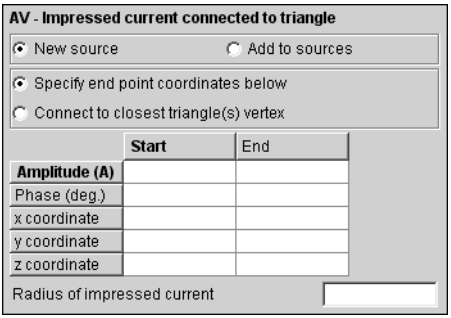


Figure 10-15: 3D radiation pattern of a spherical TE-mode with $n = 5$ and $m = 0$.

10.2.17 AV card

With this *two line* card an impressed current source is specified similar to that of the AI card, but with the AV card the end point makes electrical contact with a conducting surface as shown in figure 10-16. The current varies linearly between the value at the start point and that at the end point. At the connection point special singular functions are used for the surface current density on the triangles to allow continuous current flow.



Parameters:

New source: New excitation, replaces all previous excitations.

Add to sources: Additional excitation, is added to previous excitations.

Specify end point coordinates below: The coordinates of the end point $\vec{r}_2$ are known and specified with the *x, y, z coordinate* fields. This point must coincide with a corner point of one or more triangles.

Connect to closest triangle(s) vertex: The coordinates of the end point $\vec{r}_2$ are not known. In this case the *x, y, z coordinate* fields of the *End* point are not used. FEKO searches through all the metallic triangles for the corner point that is closest to the start point $\vec{r}_1$ of the current element. This is then the end point $\vec{r}_2$. For both cases FEKO automatically searches for all the triangles making electrical contact with the end point.

Amplitude: Current amplitude (in A) at the *Start,* $\vec{r}_1$, and *End,* $\vec{r}_2$, points.

Phase: Phase of the current at the start point in degrees.

x, y, z coordinate: Coordinates of the start and end points in m. (Note that all the coordinate values are optionally scaled by the SF card.)

Radius of impressed current: This parameter is optional. If specified, and different from zero, this value gives a finite wire radius for the impressed current element. FEKO then assumes that the current is uniformly

distributed on the wire surface and uses the exact wire integral. If the parameter is not specified, the current filament approximation is used. (This value is optionally scaled by the SF card.)

The following restrictions apply when using the impressed current elements making electrical contact with conducting surfaces.

- All the restrictions given in the discussion of the AI card also apply in this case.
- The start point of the impressed current segment may be connected with AI cards or further AV cards. If there is a current discontinuity at this point, the resulting point charge is not considered (see the discussion given with the AI card). Line charges along the current path and surface charges on the triangles are correctly taken into account. At the connection point $\vec{r}_2$ a continuous current model is used such that a point charge is not possible here.

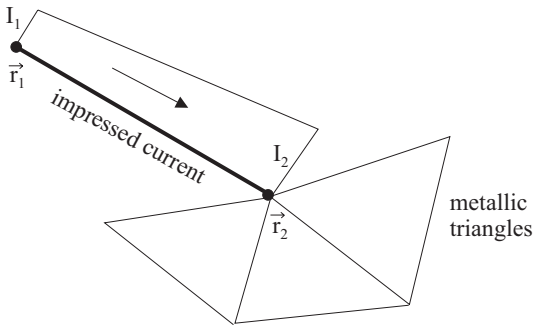
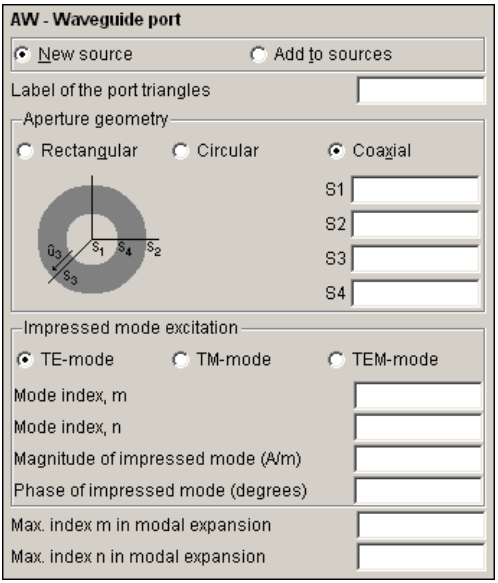


Figure 10-16: Impressed line current with a linear current distribution and electrical contact to conducting triangles

10.2.18 AW card

With this card a waveguide port excitation by an impressed mode on a rectangular, circular, or coaxial waveguide can be modelled.



Parameters:

New source: New excitation, replaces all previous excitations.

Add to sources: Additional excitation, is added to previous excitations.

Label of the port triangles: Label of the triangular patches in the mesh which represent the waveguide port.

Rectangular: A rectangular waveguide cross section is used, which is defined by three points S1, S2, and S3 as follows: S1 is an arbitrary corner point, and S2 and S3 are two corner points which define the waveguide sides $\hat{u}_1$ (from S1 to S2) and $\hat{u}_2$ (from S1 to S3). The direction in which the mode is launched is given by $\hat{u}_3 = \hat{u}_1 \times \hat{u}_2$.

Circular: A circular waveguide cross section is used. The point S1 denotes the centre of the circular port, and the point S2 specifies the radius and start point for the angular dependency. A further point S3 must be perpendicular above the centre of the circular plate, such that the direction from S1 to S3 indicates the direction in which the waveguide modes are launched.

Coaxial: Here a feed of a coaxial waveguide with circular cross sections of both the inner and outer conductor can be specified. The point definitions are the same as for the circular waveguide, except that an additional point S4 must be defined between S1 and S2 which specifies the radius of the inner conductor.

TE–mode: If this option is checked, a mode $\mathrm{TE}_{m,n}$ (also referred to as $\mathrm{H}_{m,n}$) is used as excitation.

TM–mode: If this option is checked, a mode $\mathrm{TM}_{m,n}$ (also referred to as $\mathrm{E}_{m,n}$) is used as excitation.

TEM–mode: If this option is checked (only available for the coaxial waveguide since such modes don't exist in rectangular/circular waveguides), a TEM mode is used as excitation.

Mode index m: The index m of the $\mathrm{TE}_{m,n}$ or $\mathrm{TM}_{m,n}$ mode which is impressed at the port. Note that for a rectangular waveguide the index m is related to the $\hat{u}_1$ direction (i.e. from point S1 to S2). For a circular/coaxial waveguide, m denotes the angular dependency.

Mode index n: The index n of the $\mathrm{TE}_{m,n}$ or $\mathrm{TM}_{m,n}$, mode which is impressed at the port. Note that for a rectangular waveguide the index n is related to the $\hat{u}_2$ direction (i.e. from point S1 to S3). For a circular/coaxial waveguide, n denotes the radial dependency.

Magnitude of the mode in ...: Absolute value of the complex amplitude of the impressed mode. For a TE-mode the unit is A/m, whereas for TM- or TEM-modes this is in V/m. Note that an amplitude of zero can also be specified. In this case a waveguide port is acting purely as a passive port (e.g. as waveguide termination), and no wave is launched.

Phase of the mode: The phase of the complex amplitude of the impressed mode in degrees.

Max. index m in modal expansion: At a waveguide port a specific mode is used as impressed excitation. However, due to discontinuities, also higher order modes can result and will be propagating back to the port (applies to both active and passive ports). Here the input value denotes the maximum mode index m which will be used to expand the backwards travelling waves. It must be high enough to capture the resulting field distribution. But note, the higher this value is chosen, the larger the impact on the run-time, and in particular also the mesh across the waveguide port must be chosen fine enough to represent the potential rapid field variation of higher order modes.

Max. index n in modal expansion: See description above for the max. mode index m, here similar meaning for the maximum mode index n.

In order to model a waveguide feed by an impressed mode, the cross section of the waveguide at the port location must be meshed into metallic triangles with a unique label. The normal vector of these triangular patches defines the direction in which the mode is launched; on the opposite side the impressed field of the excitation is zero. The direction of the normal vector of the mesh elements must be consistent to the orientation of the port as such, which is given by the unit vector $\hat{u}_3$, see the small graphics in the AW card panel above.

In general, specific meshing rules exist in FEKO relating the triangular patch size to the wavelength, see section 2.2.2 for details. When meshing the cross section of a waveguide to define a waveguide port, the mesh size must be small enough to capture the field distribution of the highest mode (m, n) which is included in the expansion. FEKO checks this automatically and gives a warning for coarse meshes or an error is the mesh size is too large. One must then either refine the mesh just at the port or reduce the maximum mode indices used in the expansion.

The following restrictions apply when using a waveguide port feed:

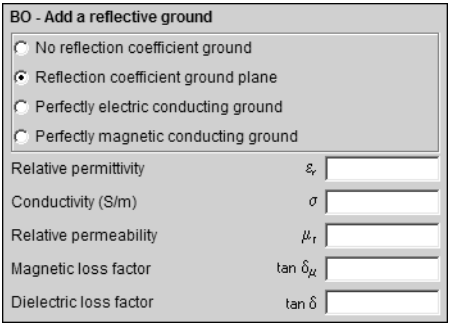
- Presently waveguide ports are available only for metallic objects (wires and surfaces and wire / surface junctions).
- Any type of dielectrics except wire coatings or thin dielectric sheets are not allowed (i.e. also no FEM or special Green's functions)
- When using waveguide ports, then UTD or PO are not allowed in the same model. Note, however, that in FEKO it is supported by means of the AR or AP cards to decompose a model (say a horn antenna in front of a reflector) into different sub-problems, and then waveguide feeds can be used. See `example_35` for an illustration of this decomposition technique.

Computing S-parameters for waveguide ports with the SP card is supported. Also multiple ports (active and/or passive) can be present in the model. S-parameters are directly based on the waveguide impedance of the specific mode under consideration. The reference impedance as specified at the SP card will not be used.

Examples for an application of waveguide feed can be found in the Examples Guide, there `waveguide_step` (S-parameter computation for a waveguide discontinuity), `example_08` (horn antenna), and `example_34` (coaxial cable).

10.2.19 BO card

With this card a ground plane (at $z = 0$) can be specified for all computations following the BO card. The reflection coefficient method is used.



Parameters:

No reflection coefficient ground: No ground plane (or use one of the other ground plane options — such as the GF card for an exact model of real ground, or the SY card for a perfect ground plane). This option is used to switch off the reflection ground if the effect of different grounds are considered in a single input file.

Reflection coefficient ground plane: Use the reflection coefficient ground plane approximation with the material parameters specified in the remaining input fields.

Perfectly electric conducting ground: Use an ideal electric ground in the plane $z = 0$. In this case the remaining parameters are ignored.

Perfectly magnetic conducting ground: Use an ideal magnetic ground in the plane $z = 0$. Also in this case the remaining parameters are ignored.

Relative permittivity: The relative dielectric constant ε_r of the ground.

Conductivity: The conductivity σ in $\frac{1}{\Omega \mathrm{m}}$ of the ground.

Relative permeability: The relative permeability μ_r of the ground.

Magnetic loss factor: Magnetic loss tangent $\tan \delta_\mu$ of the ground (the complex permeability is then given by $\mu = \mu_0 \mu_r \left(1 - j \tan \delta_\mu\right)$).

Dielectric loss factor: Electric loss tangent $\tan \delta$ (an alternative way to specify the conductivity σ — the two loss terms are related by $\tan \delta = \frac{\sigma}{\omega \varepsilon_r \varepsilon_0}$ and they have different frequency behaviour).

It should be noted that it is not possible to calculate the fields below the ground plane, i.e. it is not possible to calculate the fields in the region $z < 0$. In addition all structures must be in the region $z > 0$. If calculations inside the ground are required, for example when there are structures below ground, the exact Sommerfeld integrals (GF card) must be used.

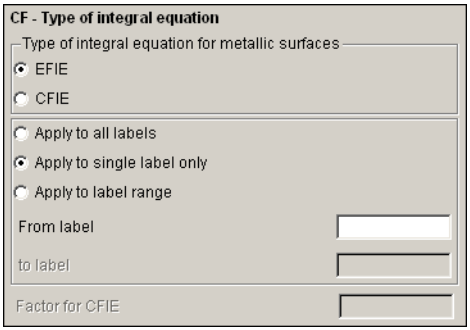
When using a perfect electric or magnetic reflection coefficient ground plane, structures can be arbitrarily close to the ground (while remaining above it). Segment end points and triangle edges lying in the plane of the ground plane will make electrical contact with a perfect electric ground plane. For a perfect magnetic ground the boundary condition forces the current to zero at this point.

If real ground parameters are used the reflection coefficient approximation is more accurate for structures further from the ground plane. Typically structures should not be closer than about $\frac{\lambda}{10}$ (FEKO will give a warning if this is the case).

A dielectric ground (real earth) can only be used with bodies treated with MoM, MLFMM, PO, FEM, or the hybrid MoM/PO, i.e. the hybrid MoM/UTD method cannot be used in the presence of a real ground.

10.2.20 CF card

Set the type of integral equation for perfectly conducting metallic surfaces.



Parameters:

Type of integral equation for metallic surfaces: Here one can chose the EFIE (*electric field integral equation*) and also the CFIE (*combined field integral equation*). See the comment below for more details.

Apply to all labels: The selected type of integral equation is applying globally to all metallic surfaces, irrespective of their label.

Apply to single label only: Here the selection of the type of integral equation applies to a single label only, which is entererd into the field *From label*.

Apply to label range: Here the selection of the type of integral equation applies to a range of labels, which is entererd into the fields *From label* and *to label*.

Factor of CFIE: In the CFIE formulation electric and magnetic terms are combined with a factor. Leaving this input field empty will select the default value of 0.2, which should normally be used.

The EFIE is the default in FEKO if no CF card is used or for labels where no setting is made at the CF card. It is the most general formulation and can be applied to both open and closed bodies. The CFIE can be used only in connection with closed objects, and the advantage is that the conditioning of the system of linear equations is better. In particular in connection with MLFMM the convergence can be improved if the CFIE is used for closed parts of an object.

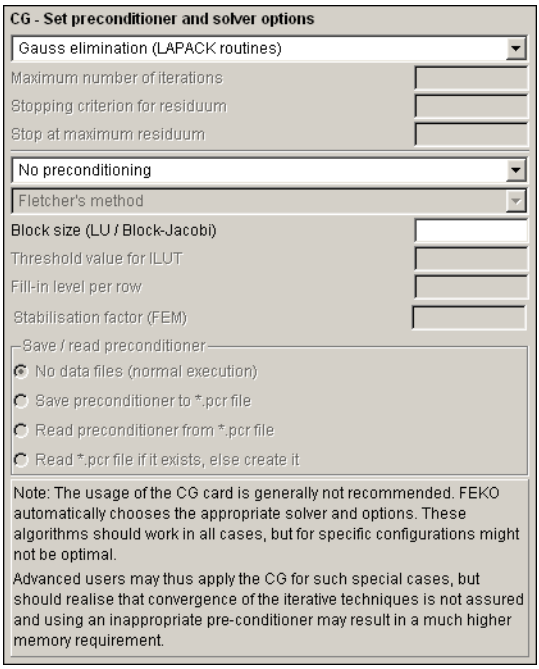
For the CFIE in addition to the fundamental restriction that the surface must be closed these further conditions apply:

- The normal vector must point outwards (i.e. from the closed field free region into the domain of interest where there are sources and fields shall be computed).
- Using symmetry in order to reduce the memory or run-time is not supported in FEKO (it will be switched off automatically).
- The CFIE formulation for the MoM cannot be used together with the MoM/PO, MoM/UTD, or MoM/FEM hybrid methods.
- The CFIE formulation can be used only with the free space Green's function (i.e. using the spherical or planar multilayer Green's functions is not supported).
- When using the CFIE one cannot have dielectric bodies in the same model, and also the CFIE surfaces must be perfectly conducting (no coating or Skin effect etc.).
- Using a ground plane (BO card) is not supported.

Note that multiple CF cards can be used in order to specify for instance that the CFIE shall be used at multiple distinct labels which do not form a range. The setting for *Factor for CFIE* is global (not per label), the value read from the last CF card will be used.

10.2.21 CG card

Here the method used to solve the matrix equation may be chosen. Normally the CG card should not be used. FEKO has been written to chose the optimal solution technique and also the corresponding pre-conditioners and options etc. for different types of problems automatically. These algorithms should work in all cases, but for specific MLFMM or FEM configurations (which rely on iterative solvers) might not be optimal. Thus for these solutions advanced users might after consultation with FEKO technical support apply the CG card in such special cases. But one should realise that convergence of the iterative techniques cannot be assured, and also when using an inappropriate preconditioner the memory requirement might be much higher than required. Also users should take care to reconsider any CG card settings in models that are derived from models containing the CG card.


CG - Set preconditioner and solver options
Gauss elimination (LAPACK routines)
Maximum number of iterations
Stopping criterion for residuum
Stop at maximum residuum
No preconditioning
Fletcher's method
Block size (LU / Block-Jacobi)
Threshold value for ILUT
Fill-in level per row
Stabilisation factor (FEM)
Save / read preconditioner
No data files (normal execution)
Save preconditioner to *.pcr file
Read preconditioner from *.pcr file
Read *.pcr file if it exists, else create it
Note: The usage of the CG card is generally not recommended. FEKO automatically chooses the appropriate solver and options. These algorithms should work in all cases, but for specific configurations might not be optimal.
Advanced users may thus apply the CG for such special cases, but should realise that convergence of the iterative techniques is not assured and using an inappropriate pre-conditioner may result in a much higher memory requirement.

Parameters:

In the first drop-down list, the matrix solution method is chosen:

- *Gauss Elimination (LINPACK routines)* Use Gauss elimination from the LINPACK routines.
- *Conjugate Gradient Method (CGM)*
- *Biconjugate gradient method (BCG)*

- *Iterative solution with band matrix decomposition*
- *Gauss elimination (LAPACK routines)* Use Gauss elimination from the LAPACK routines.
- *Block Gauss algorithm (matrix saved to disk)* The block Gauss algorithm is used (in case the matrix has to be saved on the hard disk, i.e. a sequential out-of-core solution is performed).
- *CGM (Parallel Iterative Method)*
- *BCG (Parallel Iterative Method)*
- *CGS (Parallel Iterative Method)*
- *Bi-CGSTAB (Parallel Iterative Method)*
- *RBi-CGSTAB (Parallel Iterative Method)*
- *RGMRES (Parallel Iterative Method)*
- *RGMRESEV (Parallel Iterative Method)*
- *RCGR (Parallel Iterative Method)*
- *CGNR (Parallel Iterative Method)*
- *CGNE (Parallel Iterative Method)*
- *QMR (Parallel Iterative Method)*
- *TFQMR (Parallel Iterative Method)*
- *Parallel LU-decomposition (ScaLAPACK routines)* Parallel LU-decomposition with ScaLAPACK (solution in main memory) or with out-of-core ScaLAPACK (solution with the matrix stored on hard disk). This is the default option for parallel solutions and normally the user need not change it.
- *QMR (QMRPACK routines)*
- *Direct sparse solver* Direct solution method for the FEM (no preconditioning).

Maximum number of iterations: The number of iterations limit for the iterative techniques.

Stopping criterium for residuum: Termination criterion for the normalised residue when using iterative methods. (Terminate with convergence when the normalised residue is smaller than this value.)

Stop at maximum residuum: For parallel iterative methods, the solution is terminated when the residuum becomes larger than this value. (Terminate with divergence when the normalised residue is larger than this value.)

Preconditioners: In this list the preconditioner is selected from:

- *No preconditioning:* No preconditioning is used.
- *Scaling the matrix A:* Scaling the matrix $[A]$, so that the elements on the main diagonal are all normalised to one.

- *Scaling the matrix* $[A]^H\,[A]$*:* Scaling the matrix $[A]^H\,[A]$, so that the elements on the main diagonal are all normalised to one.
- *Block-Jacobi preconditioning using inverses:* The inverses of the preconditioner are calculated and applied during every iteration step. For performance reasons *Block-Jacobi preconditioning using LU-decomposition* is recommended.
- *Neumann polynomial preconditioning:* Self explanatory.
- *Block-Jacobi preconditioning using LU-decomposition:* Block-Jacobi preconditioning where for each block a LU-decomposition is computed in advance, and during the iterations a fast backward substitution is applied.
- *Incomplete LU-decomposition:* Use an incomplete LU-decomposition of the matrix as a preconditioner.
- *Block-Jacobi preconditioning of MLFMM one-level-up:* Special preconditioner for the MLFMM, where additional information is included into the preconditioner.
- *LU decomposition of FEM matrix:* An LU decomposition of the FEM matrix is used as preconditioner.
- *ILUT decomposition of FEM matrix::* An incomplete LU decomposition with thresholding of the FEM matrix is used as preconditioner.
- *Multilevel ILUT/Diagonal decomposition of the FEM matrix:* Self explanatory.
- *Multilevel ILUT/ILUT decomposition of the FEM matrix:* Self explanatory.
- *Multilevel LU/Diagonal decomposition of the FEM matrix:* Self explanatory.
- *Sparse Approximate Inverse (SPAI) preconditioner:* Preconditioner which can be used in connection with the MLFMM (for parallel MLFMM runs this is the only supported preconditioner)

Options for the BCG: This applies to the Biconjugate Gradient Method. Options are:

- *Fletcher's method:*
- *Jacob's method:*
- *Fletcher's method, pre-iteration using Fletcher's method:*
- *Fletcher's method, pre-iteration using Jacobs' method:*
- *Jabobs' method, pre-iteration using Fletcher's method:*

Block size: The block size to be used for LU-decomposition with LAPACK as well as for the Block-Jacobi preconditioning. When nothing is specified (i.e. the input field is left empty), then appropriate standard values are used for LAPACK and the block preconditioners.

Threshold value for ILUT: This is the thresholding value used for the FEM in connection with the ILUT preconditioners.

Fill-in level per row: This is used by the MLFMM or FEM during the iterative matrix solution in connection with incomplete LU preconditioners.

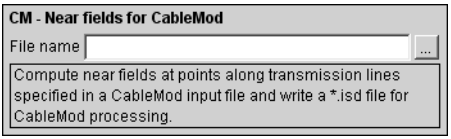
Stabilisation factor (FEM): This applies only to the incomplete LU preconditioners of the FEM and can be used to get better convergence for the FEM in critical cases (the value range is between 0 and 1).

Save/read preconditioner: For the incomplete LU preconditioners used with the FEM one has here the option to save run-time by computing the preconditioner only once and write to a `*.pcr` file and then for a subsequent run just read again from this file. Since the FEM preconditioners depend only on the FEM matrix even for combined MoM/FEM hybrid methods, this method is useful for instance when only the MoM part of a problem setup has changed.

Iterative solutions are used for instance for the MLFMM or the FEM. There might then be situations where during the iterations the residuum is decreasing but not reaching the specified stopping criterion. Instead of waiting very long until the maximum number of iterations has been reached, the user can press Ctrl-C or Ctrl-Break (under Windows) or send the SIGINT/SIGTERM signals (under UNIX) so that FEKO will stop with the iterations and resume with the further processing (e.g. far-field computations) using the solution associated with the best residuum obtained so far. To really interrupt a FEKO job Ctrl-C or Ctrl-Break must be pressed a second time (or the corresponding signal must be sent once more).

10.2.22 CM card

This card is used to couple FEKO with the transmission line simulation program CableMod or the PCB tool PCBMod[9] to calculate the coupling of electromagnetic fields into transmission lines. (The AC card is used for the case of radiation by these lines.)


CM - Near fields for CableMod
File name
...
Compute near fields at points along transmission lines specified in a CableMod input file and write a *.isd file for CableMod processing.

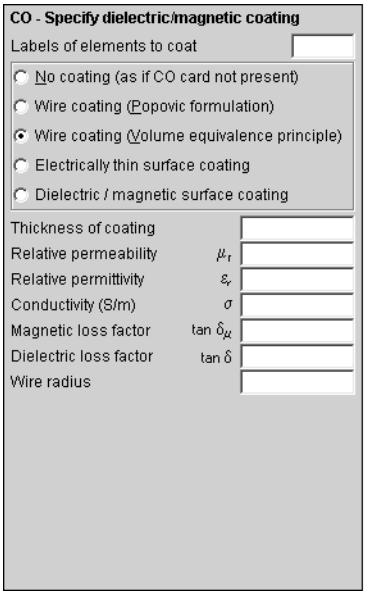
The only parameter of this card is the file name of a `*.rsd` file created by CableMod or PCBMod (enclosed in double quotation marks and starting at or after column 91). The `*.rsd` file contains geometry of the line. With the CM card FEKO calculates the electric and magnetic near field at points along the line and write these to a `*.isd` file for further processing by CableMod or PCBMod. (The `*.isd` file also contains additional data required by CableMod or PCBMod, for example the frequencies that were used during the solution.)

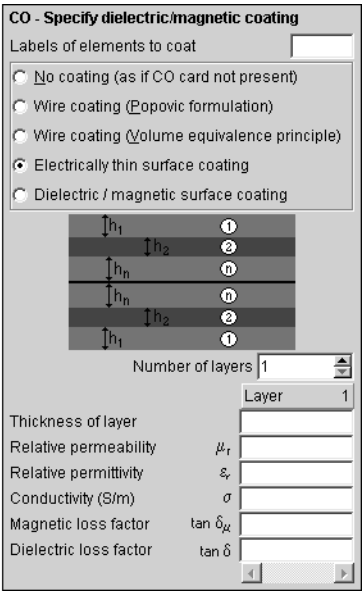
The complete geometry (without the transmission line) as well as the frequency and excitation (Ax cards) must be defined in FEKO.

[9] To use this the CableMod/PCBMod module must be activated, if required please contact your distributor.

10.2.23 CO card

This card specifies a dielectric or magnetic coating of wire segments or triangular surface elements. The coating applies to all calculations following the CO card.


CO - Specify dielectric/magnetic coating
Labels of elements to coat
No coating (as if CO card not present)
Wire coating (Popovic formulation)
Wire coating (Volume equivalence principle)
Electrically thin surface coating
Dielectric / magnetic surface coating
Thickness of coating
Relative permeability μ_r
Relative permittivity ε_r
Conductivity (S/m) σ
Magnetic loss factor $\tan\delta_\mu$
Dielectric loss factor $\tan\delta$
Wire radius



Parameters:

Label of elements to coat: All segments or triangles with this label are coated.

No coating: No coating present (as if the relevant label has no CO card). This is used to remove wire coatings from earlier solutions.

Wire coating (Popovic formulation): In this case, the radius of the metallic core is changed internally to model the change in the capacitive loading of the wire and a corresponding inductive loading is added. The only restriction of this method is that the loss tangents of the wire coating and of the surrounding medium must be identical (for instance both media could be lossless).

Wire coating (volume equivalence principle): Here the radius of the metallic wire is retained. The effect of the dielectric layer is accounted for by a volume polarisation current. The only restriction of this method is that the layer may not be magnetic in nature (i.e. the relative permeability μ_r of the coating must be the same as that of the surrounding medium).

Electrically thin surface coating: This option adds multilayer dielectric / magnetic coatings to the surface triangles with the specified label. The layers may have different permittivity and permeability, but the total coating must be electrically (i.e. relative to the wavelength in the coating) as well as geometrically thin (see the next item).

Dielectric / magnetic surface coating: This option adds electrically thick multilayer dielectric / magnetic coatings to the surface triangles with the specified label. Here it is only required that the total coating must be geometrically thin, i.e. it must be thin relative to the triangle size (and thus also to the free space wavelength) as well as the radius of curvature of the surface. This option may only be applied to elements treated with PO.

Number of layers: This field is only applicable for surface coatings. Note that the layers are symmetric around the conductor and that the layer assigned the highest number is closest to the conductor.

Thickness of . . . : For surface coatings, the thickness h_n of each respective layer; for wire coatings this is the radius of the coating less the radius ϱ of the wire-core. This value is in m and is scaled by the SF card.

Relative permeability: Relative permeability μ_r of the coating / layer.

Relative permittivity: Relative permittivity ε_r of the coating / layer.

Conductivity: Conductivity σ in $\frac{1}{\Omega \mathrm{m}}$ of the outer layer.

Magnetic loss factor: Magnetic loss tangent $\tan\delta_\mu$ of the coating / layer (the complex permeability is then given by $\mu = \mu_0\mu_r\,(1 - j\tan\delta_\mu)$).

Dielectric loss factor: Dielectric loss tangent $\tan\delta$ of the coating / layer (an alternative way to specify the conductivity σ, the two loss terms are related by $\tan\delta = \frac{\sigma}{\omega\varepsilon_r\varepsilon_0}$ and have different frequency behaviour).

Wire radius: Wire radius ϱ of the metallic wire, without layers, in m (it is scaled by the SF card). This overrides the values specified with the IP card. This field is only applicable to wire coatings.

When using the Popovic formulation for wire coatings, the following restrictions apply:

- The loss tangent $\tan\delta$ of the layer (which is calculated from the conductivity σ and the relation $\tan\delta = \frac{\sigma}{\omega\epsilon_r\epsilon_0}$) has to be identical to the loss tangent of the surrounding medium (specified with the EG card, usually free space)
- Due to the change in the radius of the metallic core, no SK card should be active for the same label, otherwise the skin effect and/or the ohmic losses refers to the wire with changed radius.

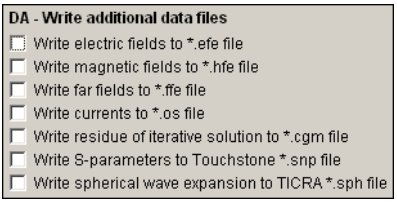
- For pure dielectric layers (i.e. the relative permeability μ_r of the layer equals that of the surrounding medium) the option *Wire coating (Volume equivalence principle)* is recommended.

Note that for wire coatings, no surface triangles with the same label are allowed. Likewise for surface coatings, no segments with same label are allowed.

If *Dielectric / magnetic surface coating* (the electrically thick coating) is used, it must remain consistent for the whole FEKO run. (It cannot be on for one solution and off for the next.) It is, however, allowed to change the thickness and the medium parameters of the coating.

10.2.24 DA card

With this card some data like near fields or S-parameters can be exported to additional ASCII files. The card allows to switch this export on and off, and affects only cards for the computation of, for example, near fields or S-parameters that follow the DA card. By default no export files are created.


DA - Write additional data files
Write electric fields to *.efe file
Write magnetic fields to *.hfe file
Write far fields to *.ffe file
Write currents to *.os file
Write residue of iterative solution to *.cgm file
Write S-parameters to Touchstone *.snp file
Write spherical wave expansion to TICRA *.sph file

Parameters:

Write electric fields . . . : The electric field strength is stored in a `*.efe` file.

Write magnetic fields . . . : The magnetic field strength is stored in a `*.hfe` file.

Write far fields . . . : The far field is stored in a `*.ffe` file.

Write currents . . . : The currents are stored in a `*.os` file.

Log residue . . . : The residue from the iterative algorithm used to solve the matrix equation is stored in a `*.cgm` file.

Write S-parameters . . . : The S-parameters (see the SP card) are written to a file in Touchstone `*.SnP` format. The `n` here gives the number of ports.

Write spherical wave expansion . . . : A spherical wave expansion of the far-field as computed by FEKO is exported to an SWE file (extension `*.sph`) which can be imported into GRASP from TICRA (code for reflector antenna modelling). Note that the far-field computation with spherical wave expansion must be requested subsequently (see the FF card in section 10.2.28).

More than one DA card is allowed in one input file. Thus, using the following sequence of control cards, with the appropriate options only certain blocks will be saved to the data files:

```
DA  ...      ** Write near fields on
FE  ...
DA  ...      ** Write near fields off
FE  ...
```

With this sequence, the electric fields calculated with the first FE card can be written to the `*.efe` file, but not those of the second FE card. The structure of the data files is described below:

***.efe file** When calculating the electric near field in Cartesian coordinates, there are columns with the position in x,y and z as well as the field components E_x, E_y and E_z in complex form. In cylindrical coordinates the columns consist of the following r, φ, z, E_r, E_φ and E_z. In spherical coordinates the columns consist of the following r, ϑ, φ, E_r, E_ϑ, E_φ.

***.hfe file** Same form as the `*.efe` file, except that the magnetic field values are given.

***.ffe file** Here the angles ϑ and φ are given. Following them are the field components E_ϑ and E_φ in complex form, with the factor $\frac{e^{-j\beta_0 R}}{R}$ left out as in the output file. The gain separated into the polarisation directions or the radar cross section, is given in dB. See also detailed discussion of the far-field quantities in section 14.6.

***.os file** First the triangle numbers and the centroids x,y and z are given as well as the complex current densities J_x, J_y and J_z at the centroid. Then there are three values which give the absolute value of the current density at the three corner points, as averaged over all triangles, that are adjacent to the corner points. The next three complex values are the components of the complex current density vector $\vec{J}$ for the first corner point of the triangle. The following groups of three are the values for the second and third corner points of the triangles.

After the current values in the triangles, the columns containing the data for the segments follow. For each segment a segment number is given, the centroid x,y and z as well as the current I_x, I_y and I_z flowing in the segment.

***.cgm file** In this file the number of iterations is given and the resulting residue from the iterative solving process of the matrix equation.

***.SnP file** The Touchstone S-parameter filename contains the number of ports in the model. The extension is `*.s1p` for a 1-port, `*.s2p` for a 2-port and so on. For 10-port and larger structures the `p` is dropped, for example `*.s12` for a 12-port. The file contains a header (following the `#` character) which specifies the frequency unit, the parameter type, the data format and the normalising impedance for all the ports. This is followed by the data lines (which may be repeated for multiple frequencies):

1-port:	f,	$\lvert S_{11}\rvert$, $\angle S_{11}$			
2-port:	f,	$\lvert S_{11}\rvert$, $\angle S_{11}$,	$\lvert S_{21}\rvert$, $\angle S_{21}$,	$\lvert S_{12}\rvert$, $\angle S_{12}$,	$\lvert S_{22}\rvert$, $\angle S_{22}$
3-port:	f,	$\lvert S_{11}\rvert$, $\angle S_{11}$,	$\lvert S_{12}\rvert$, $\angle S_{12}$,	$\lvert S_{13}\rvert$, $\angle S_{13}$	
		$\lvert S_{21}\rvert$, $\angle S_{21}$,	$\lvert S_{22}\rvert$, $\angle S_{22}$,	$\lvert S_{23}\rvert$, $\angle S_{23}$	
		$\lvert S_{31}\rvert$, $\angle S_{31}$	$\lvert S_{32}\rvert$, $\angle S_{32}$,	$\lvert S_{33}\rvert$, $\angle S_{33}$	
4-port:	f,	$\lvert S_{11}\rvert$, $\angle S_{11}$,	$\lvert S_{12}\rvert$, $\angle S_{12}$,	$\lvert S_{13}\rvert$, $\angle S_{13}$,	$\lvert S_{14}\rvert$, $\angle S_{14}$
		$\lvert S_{21}\rvert$, $\angle S_{21}$,	$\lvert S_{22}\rvert$, $\angle S_{22}$,	$\lvert S_{23}\rvert$, $\angle S_{23}$,	$\lvert S_{24}\rvert$, $\angle S_{24}$
		$\lvert S_{31}\rvert$, $\angle S_{31}$,	$\lvert S_{32}\rvert$, $\angle S_{32}$,	$\lvert S_{33}\rvert$, $\angle S_{33}$,	$\lvert S_{34}\rvert$, $\angle S_{34}$
		$\lvert S_{41}\rvert$, $\angle S_{41}$,	$\lvert S_{42}\rvert$, $\angle S_{42}$,	$\lvert S_{43}\rvert$, $\angle S_{43}$,	$\lvert S_{44}\rvert$, $\angle S_{44}$

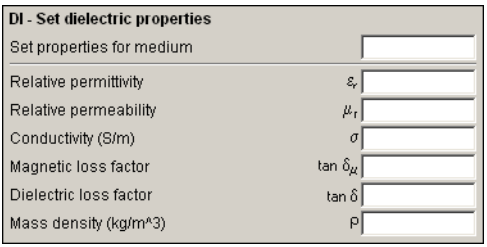
where $|S_{11}|$ is the absolute value and $\angle S_{11}$ the phase (in degrees) of the given parameter. Note that the 2-port file is formatted on a single line and in a different order than for cases with more ports.

***.sph file** This file is using the native SWE file format of TICRA as used in their code GRASP.

For all files except `*.SnP` and `*.sph` the data is in rows, i.e. each new set of data is in a new row. Complex numbers are given in the normal FORTRAN format `(Real,Imaginary)` output, e.g. `(-6.956E-03,1.034539E-07)`.

10.2.25 DI card

Here the material characteristics of the dielectric and/or magnetic bodies are entered using a medium number or index as set before at the ME card or as imported. The DI card is used for the MoM/MLFMM when using the surface current or volume current methods, or also for the FEM.



Parameters:

Set properties for medium: This gives the name of the medium as used in the ME card. The name 0 (number zero) is reserved for the surrounding free space medium.

Relative permittivity: Relative dielectric constant ε_r of the medium specified in *Set properties for medium number*.

Relative permeability: Relative permeability μ_r of the medium specified in *Set properties for medium number*.

Conductivity: Conductivity σ in $\frac{1}{\Omega \mathrm{m}}$ of the medium specified in *Set properties for medium number*.

Magnetic loss factor: Magnetic loss tangent $\tan\delta_\mu$ of the medium specified in *Set properties for medium number* (the complex permeability is then given by $\mu = \mu_0\mu_r\,(1 - j\tan\delta_\mu)$).

Dielectric loss factor: Electric loss tangent $\tan\delta$ of the medium specified in *Set properties for medium number* (this is an alternative way to specify the conductivity σ — the two loss terms are related by $\tan\delta = \frac{\sigma}{\omega\varepsilon_r\varepsilon_0}$ and have different frequency behaviour).

Mass density: The mass density (in $\mathrm{kg/m^3}$). This is only required for the SAR computations (see section 10.2.41).

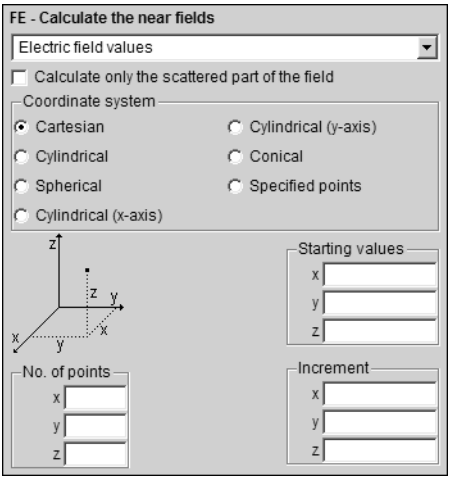
Note that for backwards compatibility with older FEKO versions (where only one medium was allowed and the parameter *Set properties for medium number* did not exist), this parameter defaults to 1 if not specified (i.e. input field is empty). If the value is set to 0 (as opposed to leaving it blank), it will overwrite the default free space parameters that might have been set at the EG card or at a previous GF card for the free space Green's function.

10.2.26 EN card

This card indicates the end of the input file. It is essential and has no parameters.

10.2.27 FE card

This card controls the calculation of the near fields.



Parameters:

First drop-down list: Here we select what to calculate:

- *No field calculation:* Field is not calculated.
- *Electric field values:* Calculate the electric field, $\vec{E}$.
- *Magnetic field values:* Calculate the magnetic field, $\vec{H}$.
- *Both electric field and magnetic field values:* Calculate both electric and magnetic fields.
- *Electric field and SAR values in cuboids:* The electric field and SAR values in the dielectric volume elements. For this option, no other parameters are not required.
- *Magnetic vector potential:* Calculate the magnetic vector potential, $\vec{A}$.
- *Gradient of the scalar electric potential:* Calculate the gradient of the scalar electric potential, $\nabla \varphi$.
- *Electric vector potential:* Calculate the electric vector potential, $\vec{F}$.
- *Gradient of the scalar magnetic potential:* Calculate the gradient of the scalar magnetic potential, $\nabla \psi$.

Calculate only the scattered part of the field: When this item is checked, only the scattered part of the field / potential is written to the output file. Otherwise the total field / potential, the sum of the scattered and

source contributions, are written to the output file. Note that depending on the used formulation in FEKO the region where the incident field as computed from the impressed sources is present might be different (for instance when using the surface equivalence formulation in the MoM to model dielectric bodies, then each source acts as incident field only in the medium where this source is located).

Coordinate system: In this dialog, the coordinate system for the calculation of the requested fields is specified.
If *Specified points* is selected here, additional dialogs for the *Number of field points* and the *Coordinates* are shown. One must then enter the coordinates of each point.
For all other cases, additional dialogs for *Starting values, Increment* and *No. of points* are shown — each with three values for the three components of the respective coordinate system.

Use old output format: If this item is checked, the old format of the near field is used in the `*.out` output file. This should only be used for compatibility with third party post processors. (POSTFEKO cannot extract SAR values from near fields in this format).

Note that all coordinates are in metres and all angles in degrees. Scaling with the SF card is only applicable when the option *Modify all dimension related values* is selected (default behaviour and highly recommended) in the SF card — in this case coordinates must be in metre after scaling.

Potentials cannot be computed with the FE card if UTD or PO is used. Also only the free space Green's function is supported - not the Green's functions for layered spheres or multilayered planar media.

If the total potentials are requested, the potentials for the sources are added. These are not available for a plane wave (A0 card) or an impressed radiation pattern (AR card) and FEKO will give an error. For a magnetic dipole (A6 card) the electric ring current model yields $\vec{A}$ and the magnetic current yields $\vec{F}$ and $\nabla\psi$; all the other potentials are zero.

If one requests `*.efe` and/or `*.hfe` files with the DA card, then $\vec{A}$ and $\nabla\varphi$ are written to the `*.efe` file, while $\vec{F}$ and $\nabla\psi$ are written to the `*.hfe` file.

If a ground plane is used, a calculation of the near fields in the ground plane is not possible. The observation points in the area $z < 0$ are not taken into account.

It should be noted that the coordinates may have an offset (OF card). Thus the near field on the surface of a sphere can be calculated, with the centre of the sphere not being located at the origin of the coordinate system.

The different *Coordinate systems* are described on the following pages.

- *Cartesian* coordinates x, y, z

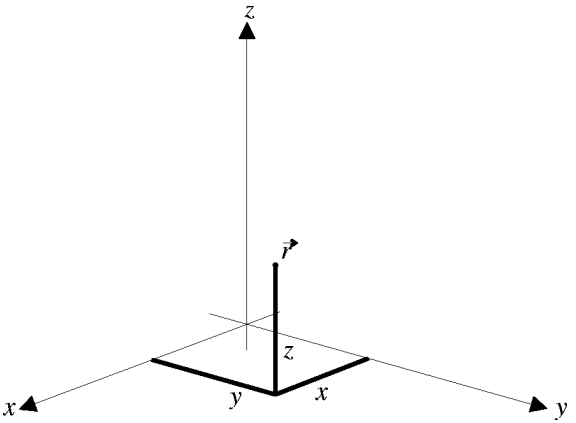


Figure 10-17: Field calculation in the *Cartesian* coordinate system

Observation Point:

$$\vec{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Unit vectors of the coordinate system:

$$\hat{x} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad \hat{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad \hat{z} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

- *Cylindrical* coordinates around z axis r, φ, z

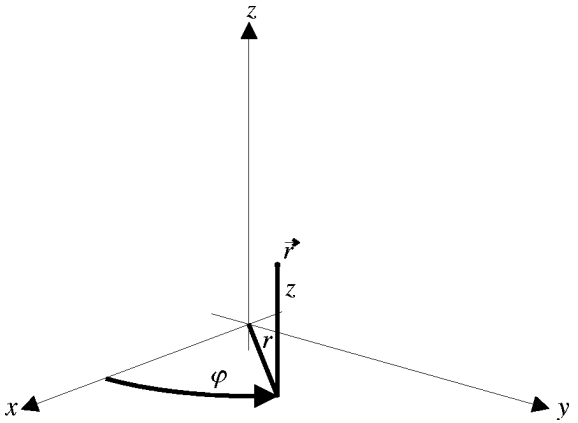


Figure 10-18: Field calculation in the *Cylindrical coordinate system*

Observation Point:

$$\vec{r} = \begin{bmatrix} r\,\cos\varphi \\ r\,\sin\varphi \\ z \end{bmatrix}$$

Unit vectors of the coordinate system:

$$\hat{r} = \begin{bmatrix} \cos\varphi \\ \sin\varphi \\ 0 \end{bmatrix} \qquad \hat{\varphi} = \begin{bmatrix} -\sin\varphi \\ \cos\varphi \\ 0 \end{bmatrix} \qquad \hat{z} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

- *Spherical* coordinates r, ϑ, φ

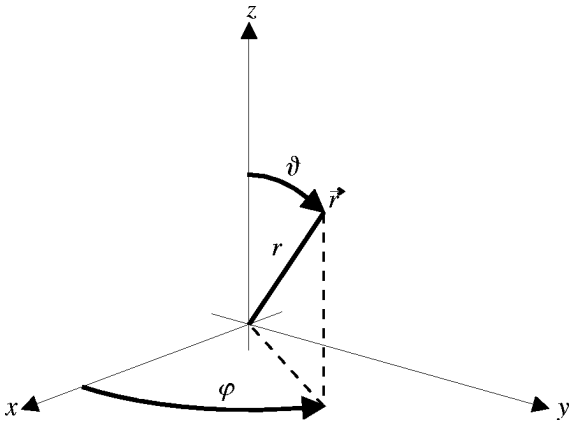


Figure 10-19: Field calculation in the *Spherical coordinate system*

Observation Point:

$$\vec{r} = \begin{bmatrix} r\,\sin\vartheta\cos\varphi \\ r\,\sin\vartheta\sin\varphi \\ r\,\cos\vartheta \end{bmatrix}$$

Unit vectors of the coordinate system:

$$\hat{r} = \begin{bmatrix} \sin\vartheta\cos\varphi \\ \sin\vartheta\sin\varphi \\ \cos\vartheta \end{bmatrix} \qquad \hat{\vartheta} = \begin{bmatrix} \cos\vartheta\cos\varphi \\ \cos\vartheta\sin\varphi \\ -\sin\vartheta \end{bmatrix} \qquad \hat{\varphi} = \begin{bmatrix} -\sin\varphi \\ \cos\varphi \\ 0 \end{bmatrix}$$

- *Cylindrical* coordinates around the x axis r, φ, x

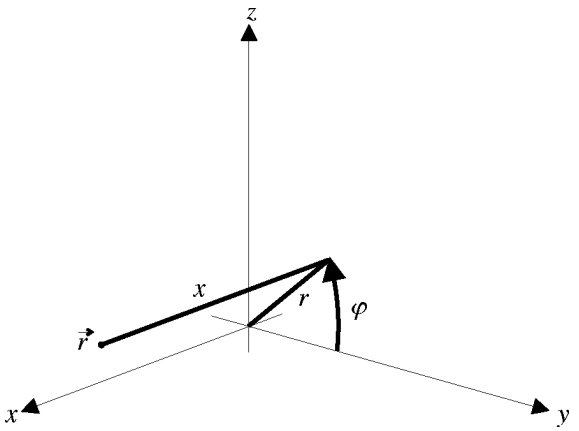


Figure 10-20: Field calculation in the *Cylindrical coordinate system around the x axis*

Observation Point:

$$\vec{r} = \begin{bmatrix} x \\ r\,\cos\varphi \\ r\,\sin\varphi \end{bmatrix}$$

Unit vectors of the coordinate system:

$$\hat{r} = \begin{bmatrix} 0 \\ \cos\varphi \\ \sin\varphi \end{bmatrix} \qquad \hat{\varphi} = \begin{bmatrix} 0 \\ -\sin\varphi \\ \cos\varphi \end{bmatrix} \qquad \hat{x} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

- *Cylindrical* coordinates around the y axis r, φ, y

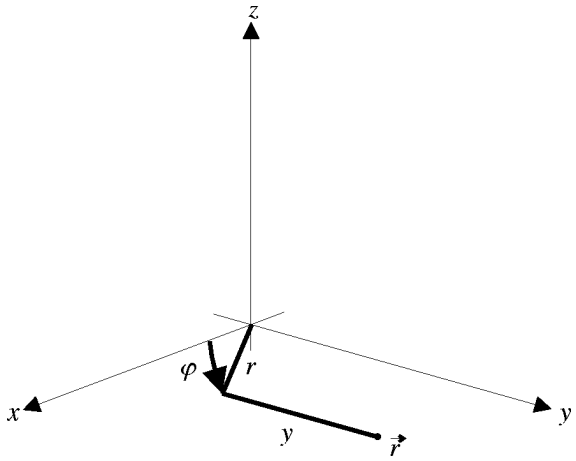


Figure 10-21: Field calculation in the *Cylindrical coordinate system around the y axis*

Observation Point:

$$\vec{r} = \begin{bmatrix} r\,\cos\varphi \\ y \\ -r\,\sin\varphi \end{bmatrix}$$

Unit vectors of the coordinate system:

$$\hat{r} = \begin{bmatrix} \cos\varphi \\ 0 \\ -\sin\varphi \end{bmatrix} \qquad \hat{\varphi} = \begin{bmatrix} -\sin\varphi \\ 0 \\ -\cos\varphi \end{bmatrix} \qquad \hat{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

- *Conical* coordinates around the z axis φ, z
  This option is similar to the field calculation in cylindrical coordinates around the z axis, where the radius r changes with the height z

$$r(z) = r_0 + \frac{\Delta r}{\Delta z} \cdot (z - z_0),$$

  and z lies within the range $z_0 \ldots z_0 + n_z \cdot \Delta z$.

  Observation Point:

$$\vec{r} = \begin{bmatrix} \left(r_0 + \frac{\Delta r}{\Delta z} \cdot (z - z_0)\right) \cdot \cos\varphi \\ \left(r_0 + \frac{\Delta r}{\Delta z} \cdot (z - z_0)\right) \cdot \sin\varphi \\ z \end{bmatrix}$$

  Unit vectors of the coordinate system:

$$\hat{r} = \begin{bmatrix} \cos\varphi \\ \sin\varphi \\ 0 \end{bmatrix} \qquad \hat{\varphi} = \begin{bmatrix} -\sin\varphi \\ \cos\varphi \\ 0 \end{bmatrix} \qquad \hat{z} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

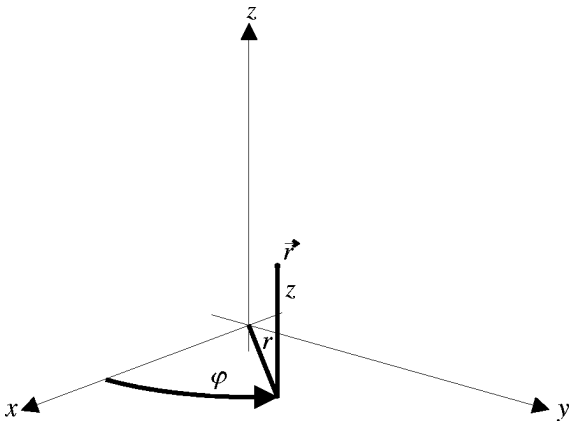
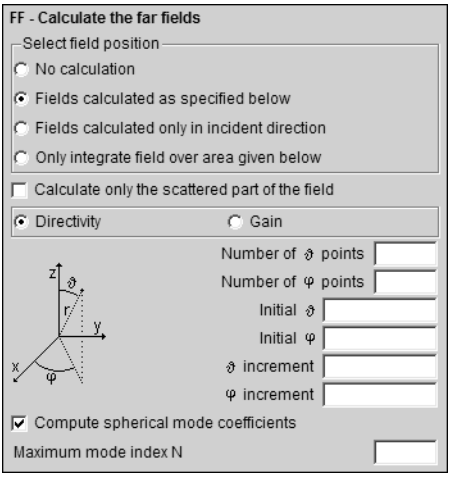


Figure 10-22: Field calculation in the *Conical coordinate system*

10.2.28 FF card

This card controls the calculation of the far fields in spherical coordinates.



Parameters:

No calculation: If this item is checked, no calculation is done.

Fields calculated as specified below: The far field is calculated with the specified settings.

Fields calculated only in the incident direction: The far field is calculated only in the incident direction (used, for example, to calculate monostatic RCS).

Only integrate field over area given below: The far field is calculated but it is not written to the output file in order to limit its size. This option is meaningful when the individual values of the field strength (such as directivity and gain) are not required, but the total radiated power has to be calculated from the integral of the Poynting vector (see the discussion below),or if one is just interested in the modal coefficients. If a `*.ffe` file has been requested with the DA card, the field values used in this integration will be written to the file.

Calculate only the scattered part of the field: When this item is checked, the field radiated by the impressed sources (such as Hertzian dipoles) is not included. This option is only meaningful if only the scattered field is required. Normally one would not check this item so that the total field is calculated. This includes all source contributions except plane wave excitations.

Directivity / Gain: Select which quantity is required.

Number of ϑ points: The number of observation points in the ϑ direction. An empty field will be set to 1.

Number of φ points: The number of observation points in the φ direction. An empty field will be set to 1.

Initial ϑ: The angle ϑ_0 in degrees of the first observation point.

Initial φ: The angle φ_0 in degrees of the first observation point.

ϑ increment: Increment $\Delta\vartheta$ in degrees of the angle ϑ.

φ increment: Increment $\Delta\varphi$ in degrees of the angle φ.

Compute spherical mode coefficients: Check this item if the spherical mode coefficients of the far field should be calculated with the FF card. The *Maximum mode index N* should be specified if this option is chosen. More information on this option is supplied below.

When calculating the monostatic radar cross section for a number of directions of incidence, the parameter *Fields calculated only in the incident direction* is necessary, otherwise *Fields calculated as specified below* can be used.

When using the FF card with *Number of ϑ points, N_ϑ*, and *Number of φ points, N_φ*, both larger than 1, the Poynting vector is integrated over the two spherical segments:

- $\vartheta_0 - \frac{1}{2}\cdot\Delta\vartheta \le \vartheta \le \vartheta_0 + (N_\vartheta - \frac{1}{2})\cdot\Delta\vartheta$ and $\varphi_0 - \frac{1}{2}\cdot\Delta\varphi \le \varphi \le \varphi_0 + (N_\varphi - \frac{1}{2})\cdot\Delta\vartheta$
- $\vartheta_0 \le \vartheta \le \vartheta_0 + (N_\vartheta - 1)\cdot\Delta\vartheta$ and $\varphi_0 \le \varphi \le \varphi_0 + (N_\varphi - 1)\cdot\Delta\varphi$

In the case of an antenna the power provided by the voltage sources must be equal to the radiated power over the whole sphere. The total radiated power can be calculated using for instance the following commands:

```
** Far field integration in angular increments of #delta (in degrees)
#delta = 5
#nt = 180 / #delta + 1
#np = 360 / #delta + 1
FF   3    #nt  #np  0          0          0          #delta    #delta
```

The output in the `*.out` file then reads for example

```
Integration of the normal component of the Poynting vector in the
angular grid DTHETA =    5.00 deg. and DPHI =    5.00 deg. ( 2701
sample points)
  angular range THETA         angular range PHI        radiated power
 -2.50 .. 182.50 deg.       -2.50 .. 362.50 deg.      5.60499E-03 Watt
  0.00 .. 180.00 deg.        0.00 .. 360.00 deg.      5.52821E-03 Watt
```

If the problem is symmetrical, it is not necessary to carry out the integration over the complete sphere. If there are three planes of symmetry (as for a simple dipole in free space) the integration only needs to be done over an eighth of the sphere. The power then has to be multiplied by 8.

If a ground plane has been specified, the calculation of the far fields below the ground plane is not possible. Observation points with $z < 0$ will thus be ignored.

While doing a far-field computation spherical mode coefficients can be computed. The calculation of these coefficients is based on the far-field values and they are written to the FEKO output file.

Doing the integration requires a two-dimensional far-field computation (i.e. both *Number of ϑ points* and *Number of φ points* larger than 1) over the whole sphere (i.e. $0 \leq \vartheta \leq 180°$ and $0 \leq \varphi \leq 360°$) and the angular increments should be chosen according to the desired accuracy and number of modes (i.e. a finer stepping is required for higher modes). It is suggested to do some initial convergence study (e.g. with increments 5° and then 1°) to see the sensitivity of the results.

Spherical modes have three indices s, m, and n with $s = 1$ for TE-modes, $s = 2$ for TM-modes, $m = -N \ldots N$, and $n = 1 \ldots N$ (see also more detailed description at the AS card in section 10.2.16, including a one-dimensional compressed indexing scheme $j = 1 \ldots J$ and the normalisation of the modes etc.). The input parameter *Maximum mode index N* determines the maximum mode index N, i.e. a total number of $J = 2N\,(N + 2)$ modes will be computed.

The modes origin is the same as for the far-field computation in general (i.e. this is the global origin unless an OF card has been used to specify an offset). It should also just be mentioned that due to the nature of the far-field (propagating towards $r = \infty$) all computed mode coefficients refer to spherical cylinder functions $z_n^{(c)}$ with type $c = 4$ (see more details at the AS card).

It shall be mentioned that when spherical modes are computed with the FF card, the DA card can also be used (must be in front of the FF card) in order to request that this spherical mode expansion is exported to an SWE file (extension `*.sph`) which can be imported into the computer code GRASP from TICRA. GRASP is a reflector antenna modelling code, and by means of this SWE file export one can for instance model a horn antenna as feed in FEKO and then export this feed structure and use in GRASP. The spherical mode expansion coefficients Q_{smn} as used by FEKO are described at the AS card in section 10.2.16. In GRASP a slightly different convention is used:

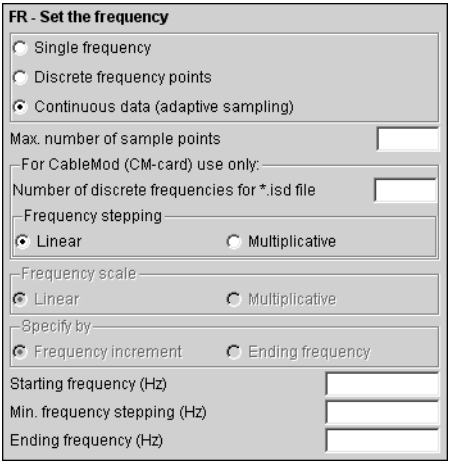
- An additional factor $\frac{1}{\sqrt{8\pi}}$ is used
- The coefficients are conjugate complex (i.e. GRASP assumes an $e^{-i\omega t}$ time dependency)

- The index m is exchanged with $-m$ (since in FEKO the φ dependency is defined differently than in GRASP, $e^{jm\varphi}$ versus $e^{-jm\varphi}$)

These conversions are done automatically by FEKO when exporting the SWE file, so that this can readily be imported into GRASP.

10.2.29 FR card

This card sets the frequency/frequencies (in Hz), at which the solution will be obtained.



The solution can be done for a single frequency, a loop of discrete frequencies (linear or multiplicative stepping), or a continuous solution in a given frequency band with adaptive frequency interpolation. For a continuous solution, only one FR card is allowed.

Parameters:

Single frequency: Only a single frequency will be analysed

Discrete frequency points: If this item is selected the following parameters are applicable:

- *Number of frequency points:* For a discrete frequency sweep, the number of frequency samples must be larger than 1.
- *Frequency scale:* In this group either *Linear* or *Multiplicative* scaling is selected. If *Linear* is selected, then consecutive frequencies differ with a fixed value, i.e. the new frequency is the previous value plus the frequency increment. If *Multiplicative* is selected, then consecutive frequencies differ by a constant factor, i.e. the new frequency is the previous value multiplied by the frequency factor.
- *Specify by:* If *Frequency increment / factor* is selected, the user specifies the increment or factor mentioned in the previous item. This, the number of frequencies and the start frequency then determine the ending frequency. If *Ending frequency* is selected, the user specifies this and the increment / factor is calculated.

Continuous data: Select this item to use an adaptive frequency interpolation technique to obtain a continuous representation of the results in the given frequency band (see Chapter 13). When using this feature, the remaining parameters have the following meaning:

- *Max number of sample points:* Maximum number of discrete frequency points in this frequency band at which FEKO may be executed (limitation to avoid convergence problems). If left empty, the default value of 1000 will be used.
- *For CableMod/PCBMod . . . , Number of discrete frequencies . . . :* This field is only relevant when the CM card is used to create an `*.isd` file. The results are written to the `*.isd` file for the number of discrete frequencies specified in this field.
- *Starting / ending frequency:* Defines the frequency range.
- *Min. frequency stepping:* Minimum increment between adaptive samples, see the note below.

In order to obtain a continuous frequency response, the adaptive frequency interpolation technique obtains the solution at a set of discrete frequency points. They are automatically placed, for example using large frequency increments in regions with a smooth behaviour of the results, and much finer frequency increments close to resonances. Sometimes, for example when using a frequency dependent mesh, the FEKO results versus frequency may contain small discontinuities. In these cases the adaptive algorithm cannot converge. (It will continue to refine the frequency increment as it tries to fit a smooth curve through the discontinuity and will only stop when the *Max. number of sample points* is reached.) One may avoid this by setting the *Frequency increment* to the minimum allowable separation distance between neighbouring frequency sample points. The value of the *Frequency increment* must be smaller than the resolution required to solve, for example, sharp resonances. If left empty, the default is

1.0E-4*(*Ending frequency* - *Starting frequency*).

If a discrete loop with more that one frequency is required then either the frequency increment or the ending frequency must be specified, but not both. If the end frequency is specified, the frequency increment is calculated from:

- for a linear frequency scale (additive increments):

$$\Delta_f = \frac{f_2 - f_1}{N_f - 1}$$

- for a multiplicative frequency scale (multiplicative increments):

$$fac = \left(\frac{f_2}{f_1}\right)^{\frac{1}{N_f - 1}}$$

where Δ_f is the frequency increment (linear stepping), fac the increment factor (multiplicative stepping), f_1 the start frequency, f_2 the ending frequency and N_f the number of frequencies.

When writing results at discrete frequencies to a `*.isd` file, the frequency increment when a linear frequency scale is used is calculated similar to the case for Δ_f as shown above.

If more than one frequency is to be examined, then all the control cards up to the next FR card or EN card will be read into a buffer and are executed for each frequency. More information can be found in section 10.1.

10.2.30 GF card

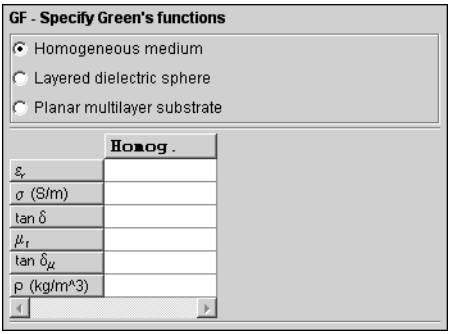
With this card the Green's function may be selected. The Green's function relates the fields in space to the sources present. The propagation space is usually free space, and FEKO uses the free space Green's function by default. For a few specific interaction environments, some of the complexities of the environments can be taken into account using special Green's functions. The Green's functions that FEKO supports are:

- *Homogeneous medium:* The dielectric properties for the entire problem space can be set. This is useful for modelling in a homogeneous medium that differs from free space.
- *Layered dielectric sphere:* A layered dielectric sphere located at the origin is taken into account with the Green's function.
- *Planar multilayer substrate:* A multilayer dielectric substrate in the xy-plane is taken into account. The substrate can have metallic ground planes above and below it.

Using Green's functions to model the presence of these dielectric regions means that their influence is taken into account implicitly, using less computational resources than modelling them using either the volume- or surface equivalence principles. The disadvantage is that they are only available for these specific environments.

The dialogs for each of the three cases above are discussed separately below.

10.2.30.1 Homogeneous medium



When this Green's function is selected, the EM problem under investigation is modelled inside an infinite space of the homogeneous medium defined in this card. This is the standard "free space" Green's function similar to when the GF card is not used. The medium is normally free space, but different parameters can be set with the EG card or in the GF card. The parameters as set by the EG card is retained only when all the real parameters except the density ρ of the GF card is empty or zero. If any of the medium parameters below is set, *all* the EG card values are overridden. (Those parameters that are not specified will then default to the values given here.)

Parameters:

ε_r*:* Relative permittivity of the homogeneous medium (if this field is empty, $\varepsilon_r = 1$ is used).

σ*:* Conductivity σ in $\frac{1}{\Omega \mathrm{m}}$ of the homogeneous medium.

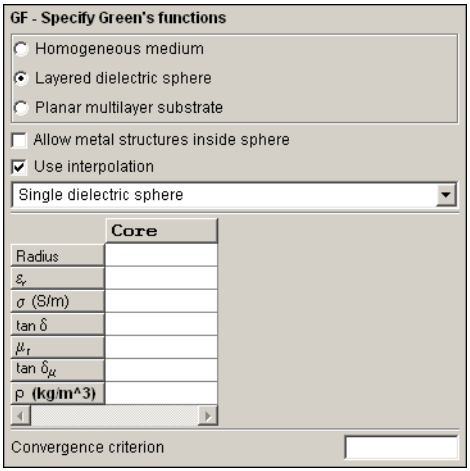
$\tan\delta$*:* Electric loss tangent $\tan\delta$ of the homogeneous medium. This is an alternative way to specify the conductivity σ — the two loss terms are related by $\tan\delta = \frac{\sigma}{\omega\varepsilon_r\varepsilon_0}$ but have different frequency behaviour. (If both $\tan\delta$ and σ *in* $\frac{1}{\Omega \mathrm{m}}$ are empty or zero, $\tan\delta$ is set to zero).

μ_r*:* Relative permeability μ_r of the homogeneous medium (if this field is empty, $\mu_r = 1$ is used).

$\tan\delta_\mu$*:* Magnetic loss tangent $\tan\delta_\mu$ of the homogeneous medium (the complex permeability is then given by $\mu = \mu_0\mu_r\,(1 - j\tan\delta_\mu)$; $\tan\delta_\mu$ is set to zero if this field is empty).

ρ*:* The mass density (in $\mathrm{kg/m^3}$). This parameter is only required for SAR computations (see section 10.2.41).

10.2.30.2 Layered dielectric sphere



When this Green's function is selected, the EM interaction of a layered dielectric sphere located at the coordinate system centre is taken into account. With this option it is, for example, possible to analyse a cellphone in front of a spherical shell model of the human head very efficiently.

Parameters:

Configuration list: The drop-down list allows selecting between *Single dielectric sphere* and a core with a number of layers. Note that whether metal structures are allowed influences the number of layers that are allowed.

Allow metal structures inside sphere: When this item is checked metallic structures can be present in the inner parts of the sphere.

Use interpolation: When this item is checked interpolation (`*.gfe` and `*.gfh` files) is used to accelerate the computations.

Convergence criterion: Convergence criteria for the summation of the rows of Green's functions. If this field is 0 or undefined, a sensible standard criterion is used.

Radius: Radius of the sphere / layer in m (is scaled by the SF card). For the layers, this is the total radius of the core plus layers up to that point. The highest numbered layer is outside.

ε_r*:* Relative permittivity of the core / layer (if this field is empty, $\varepsilon_r = 1$ is used).

σ:	Conductivity σ in $\frac{1}{\Omega \mathrm{m}}$ of the core / layer.
$\tan\delta$:	Electric loss tangent $\tan\delta$ of the core / layer. This is an alternative way to specify the conductivity σ — the two loss terms are related by $\tan\delta = \frac{\sigma}{\omega\varepsilon_r\varepsilon_0}$ but have different frequency behaviour. (If both $\tan\delta$ and σ *in* $\frac{1}{\Omega \mathrm{m}}$ are empty or zero, $\tan\delta$ is set to zero).
μ_r:	Relative permeability μ_r of the core / layer (if this field is empty, $\mu_r = 1$ is used).
$\tan\delta_\mu$:	Magnetic loss tangent $\tan\delta_\mu$ of the core / layer (the complex permeability is then given by $\mu = \mu_0\mu_r\,(1 - j\tan\delta_\mu)$; $\tan\delta_\mu$ is set to zero if this field is empty).
ρ:	The mass density (in $\mathrm{kg/m^3}$). This parameter is only required for SAR computations (see section 10.2.41).

The scaling factor that is entered by the SF card is applied to the radius. The parameters of the medium outside the sphere (usually free space) can be set with the EG card.

The Green's function for a homogeneous or layered dielectric sphere can be used with metallic structures (treated with the MoM) either inside or outside the sphere (but not for example a wire from inside to outside). It can be used with dielectric bodies treated with the volume equivalence principle (e.g. the hand of a user around a mobile phone), but the dielectric bodies must be outside the sphere.

An example of the use of the GF card for a sphere is given in `example_15` (*Examples Guide*).

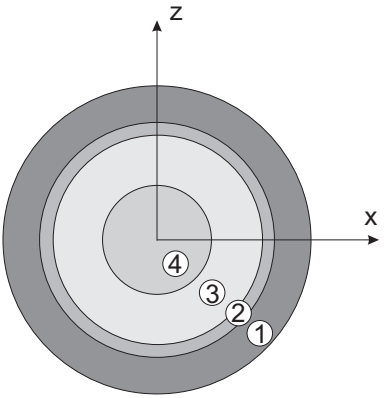
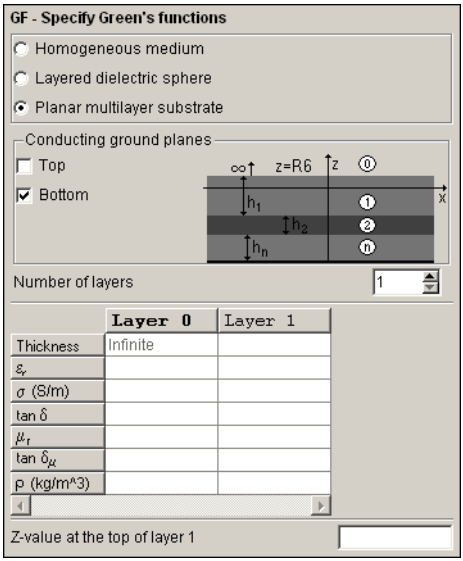


Figure 10-23: Example of a sphere consisting of 4 media (core and 3 layers) indicating the layer numbering.

10.2.30.3 Planar multilayer substrate



When this Green's function is selected, the EM interaction of a layered dielectric substrate located in the xy-plane is taken into account. The Green's function can also take the effect of a top and/or bottom ground plane into account. This formulation has been popularised by its application to planar (microstrip) circuits and antennas, but it is applicable to a larger class of EM problems e.g. buried antennas.

Parameters:

Conducting ground planes: By checking *Top* and/or *Bottom* the user can choose whether to have ground planes at the top and/or bottom of the multilayer substrate. If a ground plane is present, metallic structures and field computations are not allowed above the top or below the bottom ground plane. If a top ground plane is added, the parameters of layer 0 (the upper half-space) are ignored. If there is no ground plane at the bottom, the last layer is infinitely thick.

Number of layers: Number of layers in the substrate (layer 0 — the upper half-space) is not included in the number.

Thickness: Thickness of the layer (is scaled by the SF card).

ε_r*:* Relative permittivity ε_r of the layer.

σ*:* Conductivity σ in $\frac{1}{\Omega \mathrm{m}}$ of the layer.

tanδ: Electric loss tangent tanδ in the layer.

μ_r: Relative permeability μ_r of the layer.

$\tan\delta_\mu$: Magnetic loss tangent $\tan\delta_\mu$ in the layer (the complex permeability is given by $\mu = \mu_0\mu_r\,(1 - j\tan\delta_\mu)$).

ρ: The mass density (in kg/m^3). This parameter is only required for SAR computations (see section 10.2.41).

Z-value at the top of layer 1: The value of the z coordinate at the transition between layer 0 and layer 1.

As indicated in the figure of the card, layer 0 is the upper half-space, layer 1 is just beneath this, etc.

Losses in any layer may be specified through assignment of either the conductivity σ or by the electric loss tangent $\tan\delta$. Only one of these input values may be used; the other must remain empty. They are related by $\tan\delta = \frac{\sigma}{\omega\varepsilon_r\varepsilon_0}$. It should be noted that, inside a frequency loop with varying ω, the first option is constant while second one is not.

Metallic structures can have an arbitrary orientation (horizontal, vertical, and also diagonal). They can lie at an arbitrary position in one or more layers and can lie directly on the border of two layers (e.g. on the surface of a substrate). The only restriction is that no metallic segment or triangle may cross a boundary between layers, i.e. it must lie completely within one layer or at the boundary between layers.

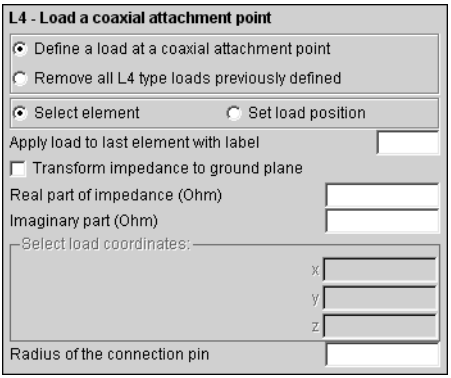
If, for example, a metallic wire penetrates a multilayer substrate, the segmentation must be such that there is a node on each interface between layers. (See `example_31` in the *Examples Guide*.)

The following is not possible with this Green's function:

- dielectric ground (BO Card)
- hybrid MoM/PO method
- hybrid MoM/UTD method
- hybrid MoM/FEM method
- dielectric bodies with the surface or volume equivalence principle

10.2.31 L4 card

This card can be used to add a load between a metallic triangle and the ground plane for the planar multilayer Green's function without having the requirement to model a vertical current element (analogous to the A4 excitation card).


L4 - Load a coaxial attachment point
Define a load at a coaxial attachment point
Remove all L4 type loads previously defined
Select element
Set load position
Apply load to last element with label
Transform impedance to ground plane
Real part of impedance (Ohm)
Imaginary part (Ohm)
Select load coordinates:
x
y
z
Radius of the connection pin

Parameters:

Define a load at a coaxial attachment point: Define a load with the following parameters.

Remove all L4 type loads previously defined: This L4 card does not define a load, but rather all previously defined L4 loads are deleted. All the other input parameters of this card are ignored.

Select element: The label of the triangle to load. If there is more than one triangle with this label, the one with the highest element number is loaded.

Set source position: Alternatively, the user may specify the Cartesian coordinates x, y and z of the load point in the *Select source coordinates* group. The triangle with the centre point closest to this point is loaded.

Transform impedance to ground plane: The specified impedance refers to the metallic ground plane, and a transformation must be done to get the correct load impedance at the triangle.

Real part of impedance: Real part of the load impedance (in Ω).

Imaginary part: Imaginary part of the load impedance (in Ω).

Radius of the connection pin: The radius of the load pin in m.

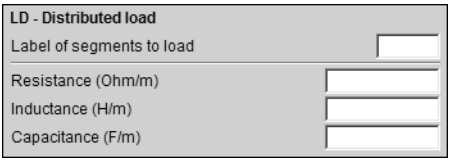
The source coordinates (if entered) and the radius of the connection pin (if entered) is optionally scaled by the SF card.

The implementation is such that if an L4 card is processed any existing L4 load on that triangle is replaced. For example, if two L4 cards are used directly below each other, the first specifying a 50 Ω and the second a 20 Ω load, the segment will be loaded with 20 Ω, not 70 Ω.

It must also be noted that if the L4 card is used in conjunction with an A4 card (impressed current source), the load impedance of the L4 card is placed parallel to the input impedance of the A4 source (it has no effect if it is in series with the current source), i.e. the resulting input admittance is the sum of input admittance without the L4 load, and the admittance added by the L4 card.

10.2.32 LD card

With this card it is possible to specify a distributed resistive, capacitive or inductive loading or even a series combination of these values for a segment.



Parameters:

Label of segments . . . :	All segments with this label are subjected to distributed loading.
Resistance:	The distributed resistance in $\frac{\Omega}{\mathrm{m}}$.
Inductance:	The distributed inductance in $\frac{\mathrm{H}}{\mathrm{m}}$.
Capacitance:	The distributed capacitance in $\frac{\mathrm{F}}{\mathrm{m}}$.

The combined impedance of the segment with length l is then

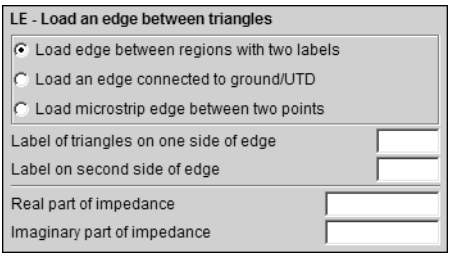
$$Z_s = \left(R' + j\omega L' + \frac{1}{j\omega C'} \right) \cdot l.$$

It should be noted that if the *Capacitance (F/m)* is left empty, it is treated as infinite, such that it does not contribute to the impedance.

The LD card may be combined with the LP, LS, LZ and the SK cards, but only one LD card may be used per label. If a second LD card is used, it replaces the values entered by the first one. This card has no significance for surface elements, even when these are assigned the same label.

10.2.33 LE card

With the LE card an edge between surface triangles can be loaded with an impedance $Z = R + jX$, as shown in figure 10-24.


LE - Load an edge between triangles
Load edge between regions with two labels
Load an edge connected to ground/UTD
Load microstrip edge between two points
Label of triangles on one side of edge
Label on second side of edge
Real part of impedance
Imaginary part of impedance

See the AE card for the excitation of such an edge. As shown in figure 10-24 the edge can consist of several single edges — each of which should be common to triangles that have one of only two labels. One of these labels must be unique, i.e. only one triangle at each single edge should have this label. (See the AE card — section 10.2.12 — for a discussion of the allowed configurations.) Alternatively the edge can be along a connection between triangles and a polygonal UTD plate or a PEC ground plane, or it can be a microstrip feed line port. The impedance Z applies to the complete edge (i.e. all the single edges in parallel). The LE card can be combined with the AE card to specify both an impedance and a voltage source over the edge.

Parameters:

Load edge between regions with two labels: The edge between two labeled regions is loaded with a complex impedance. The following parameters apply when this item is checked:

- *Label of triangles on one side of edge:* The label on the first edge of the triangle.
- *Label on second side of edge:* The label of the second edge of the triangle.

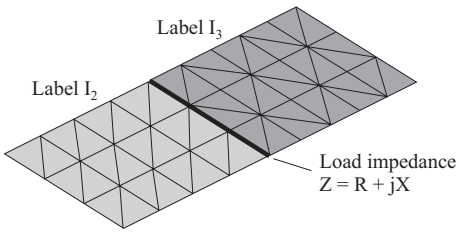


Figure 10-24: Application of the LE card

Load an edge connected to ground/UTD: Load the triangles with a specified label that are connected to a UTD surface or to a PEC ground plane (as specified with a BO or GF card). The following parameters apply when this item is checked:

- *Labels of triangles connected to ground:* The labels of the triangle that are connected to the ground.

Load microstrip edge between two points: This is a special microstrip port load. The load is placed on all edges on the line between two points (previously specified with DP cards) entered into the dialog below. A GF card with a conducting ground plane must be present. The following parameters apply when this item is checked:

- *Start point of edge:* The start point (not label) of the edge.
- *End point of edge:* The end point (not label) of the edge.

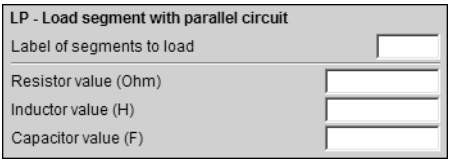
Real part of impedance: Real part of the complex load impedance.

Imaginary part of impedance: Imaginary part of the complex load impedance.

Note that the edge between the triangles does not need to be straight. One may, for example, specify a resistive connection between two half cylinders.

10.2.34 LP card

This card can be used to assign discrete circuit elements (in parallel) to a segment. Figure 10-25 shows the parallel circuit that can be assigned to a segment.



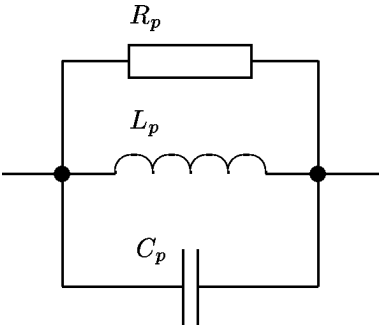


Figure 10-25: Sketch of the parallel circuit

Parameters:

Label of segments . . . :	All segments with this label are assigned the parallel circuit values specified below.
Resistor value:	Value of the resistor in Ω.
Inductor value:	Value of the inductor in H.
Capacitor value :	Value of the capacitor in F.

The impedance is then given by

$$Z_p = \frac{1}{\frac{1}{R_p} + \frac{1}{j\omega L_p} + j\omega C_p} \quad .$$

If the resistance is set to zero, then the resistance is interpreted as infinite, i.e. in the parallel case it will not change the impedance. The same applies to the inductance.

The LP card may be combined with the LD, LS, LZ and the SK cards, but only one LP card may be used per label. If a second LP card is used, it replaces the values entered by the first one. This card has no significance for surface elements, even when these are assigned the same label.

10.2.35 LS card

This card can be used to assign discrete circuit elements (in series) to a segment.

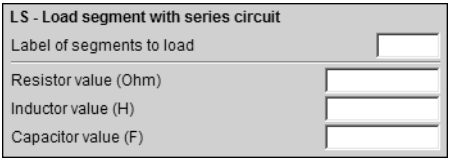


Figure 10-26 shows this circuit that can be assigned to a segment.

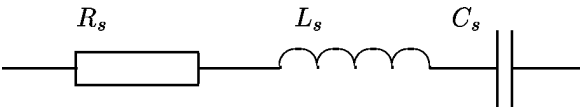


Figure 10-26: Sketch for the serial combination

Parameters:

Label of segments . . . :	All segments with this label are assigned the parallel circuit values specified below.
Resistor value:	Value of the resistor in Ω.
Inductor value:	Value of the inductor in H.
Capacitor value:	Value of the capacitor in F.

The impedance is given by

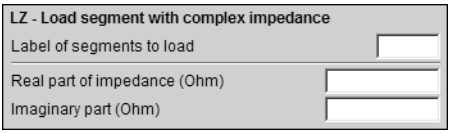
$$Z_s = R_s + j\omega L_s + \frac{1}{j\omega C_s} \quad .$$

If a capacitance of zero is selected, it is interpreted as infinite capacitance, i.e. in the case of the series combination it is zero.

The LS card may be combined with the LD, LP, LZ and the SK cards, but only one LS card may be used per label. If a second LS card is used, it replaces the values entered by the first one. This card has no significance for surface elements, even when these are assigned the same label.

10.2.36 LZ card

This card can be used to assign a complex impedance to a segment.



Parameters:

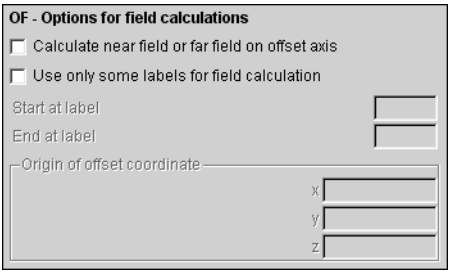
Label of segments . . . :	All segments with this label are assigned the specified impedance.
Real part of impedance:	The real part of the complex impedance in Ω.
Imaginary part:	The imaginary part value of the complex impedance in Ω.

The complex impedance value is a constant with respect to frequency. Frequency dependent impedances can be realised using the LS or the LP cards.

The LZ card may be combined with the LD, LP, LS and the SK cards, but only one LZ card may be used per label. If a second LZ card is used, it replaces the values entered by the first one. This card has no significance for surface elements, even when these are assigned the same label.

10.2.37 OF card

This card specifies an offset for the origin of the coordinate system used for near and far field calculations. In addition it is possible to use only a part of the structure when calculating the fields (selected using labels).



Parameters:

Calculate near field on offset axis: Use the offset specified below as the origin of the coordinate system for field calculations.

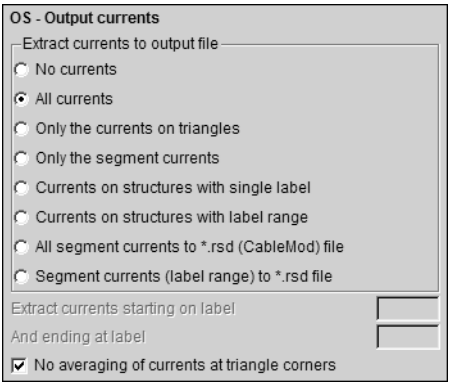
Use only some labels for field calculation: Use label selection when calculating near and far fields. Only the currents on structures with a label in the range specified in the fields *Start at label* and *End at label* are used during field computation. (If a basis function extends over, for example, two triangles it is included if either triangle of the triangles lies in the specified range.)

Origin of offset coordinate: In this group the Cartesian coordinates of the transformed origin are specified. Each of x, y and z is scaled by the SF card if the SF card is used.

A possible application of the OF card is, for example, to calculate the near field on the surface of a sphere whose centre does not lie on the origin. The OF card transforms the origin of the coordinate system to the centre of the sphere, such that the near field calculation can be executed in spherical coordinates.

10.2.38 OS card

With this card the currents on the surfaces and the segments can be extracted.


OS - Output currents
Extract currents to output file
No currents
All currents
Only the currents on triangles
Only the segment currents
Currents on structures with single label
Currents on structures with label range
All segment currents to *.rsd (CableMod) file
Segment currents (label range) to *.rsd file
Extract currents starting on label
And ending at label
No averaging of currents at triangle corners

Parameters:

No currents: No current output (but does start calculation).

All currents: Output all currents on triangles (metallic and dielectric).

Only the currents on triangles: Only output the currents on surface triangles.

Only the segment currents: Only output the currents on wires.

Currents on structures with single label: Output all currents on segments and triangles with the label *Extract currents on this label*.

Currents on structures with label range: Output all currents on segments and triangles in the label range specified by the fields *Extract currents starting on label* and *And ending at label*.

All segment currents to `*.rsd` *(CableMod) file:* Export the currents on all segments to a `*.rsd` file in CableMod/PCBMod format (see the comment below).

Segment currents (label range) to `*.rsd` *file:* Export the currents on all segments with labels in the range specified by the fields *Extract currents starting on label* and *And ending at label* to a `*.rsd` CableMod/PCBMod file.

No averaging of currents at triangle corners: For the output of currents in the vertices of the triangles, neighbouring triangles with common vertices are identified. The current densities are then averaged over the neighbours. This ensures that the graphical representation is a smooth colour representation without discontinuities. With this switch

off, the averaging that may be very time consuming, particularly for structures containing a large number of triangles, is switched off.

Multiple OS cards can be used to extract currents on multiple, non-consecutive, labels.

The options where a `*.rsd` file is written permit the creation of a `*.rsd` file for use with the transmission line simulation program CableMod or the PCB tool PCBMod[10]. The currents along all or selected segments are exported to the `*.rsd` file (the filename without extension is the same as that of the `*.fek` file). The `*.rsd` file is an ASCII file and contains first a description of the geometry of the line, followed by blocks with the current information for each frequency. It can be read by CableMod or PCBMod, and can also be imported back into FEKO to realise an impressed line source (see the AC card).

If the current of dielectric triangles (surface current formulation) must be output by the OS card, both the equivalent electric and magnetic surface currents of the external problem are written to the output file. (The currents of the internal problem are different to those of the external problem only in that their sign is reversed.)

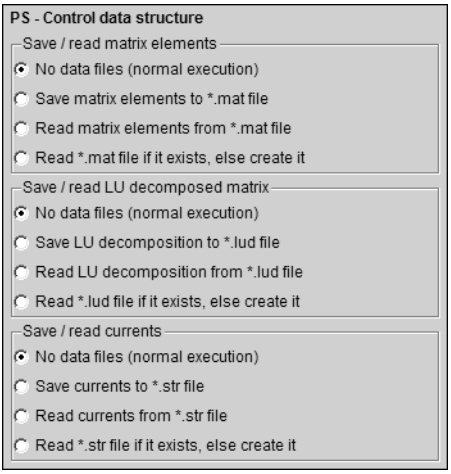
If requested by the DA card, a `*.os` card will be created in addition to the currents written to the output file.

[10]To use the CableMod/PCBMod interface this module must be activated, if required please contact your distributor.

10.2.39 PS card

This card is a general program control and can be used for instance to store the current expansion coefficients to a file or load them later again to speed up the solution.

It is important to be familiar with the solution process of the MoM to understand this card. The solution of electromagnetic problems based on the MoM involves a setting up a system of linear equations, which by default is solved using an LU-decomposition and a subsequent backwards substitution. This card can be used to save the matrix of the system of linear equations, its LU-decomposition, or the solution vector (which also includes PO currents etc.). Such elements can also be loaded again.


PS - Control data structure
Save / read matrix elements
No data files (normal execution)
Save matrix elements to *.mat file
Read matrix elements from *.mat file
Read *.mat file if it exists, else create it
Save / read LU decomposed matrix
No data files (normal execution)
Save LU decomposition to *.lud file
Read LU decomposition from *.lud file
Read *.lud file if it exists, else create it
Save / read currents
No data files (normal execution)
Save currents to *.str file
Read currents from *.str file
Read *.str file if it exists, else create it

Parameters:

Save / read matrix elements: Select this option to save or read the matrix elements of the system of linear equations. This is typically not recommended, since the file will be large and the time to recompute the matrix elements is typically much shorter than the time for the LU-decomposition of the matrix.

Save / read LU decomposed matrix: Select this option to save or read the LU-decomposition of the system of linear equations. This option is useful if you want to solve a problem subsequently with multiple different excitations (i.e. right-hand sides). Note that if you do this in one FEKO run (i.e. one `*.pre` file), then FEKO keeps the LU-decomposition automatically in memory. This option is useful if you want to solve for multiple excitations in different FEKO runs.

Save / read currents: Select this option to save or read the the solution vector of the system of linear equations. The solution vector corresponds to the currents on the structure being simulated.

FEKO always uses the most efficient computation when doing multiple solutions in one file. However, sometimes one might also do a solution, look at the results and then change certain parameters. Then the option to store the solution in the `*.str` file and load it again or the similar option for the LU-decomposition are particularly useful. The option to save the currents is applicable when the solution stays the same (i.e. same geometry, same material parameters and loads, same frequency, same excitation), but when one wants to compute the near- or far-fields with different options. The storage of the LU-decomposition is useful when only the right-hand side of the system of linear equations changes (e.g. the direction of incidence of a plane wave). FEKO checks all this using checksums and reports a warning if, for example, currents were exported for one frequency and are later imported again for another frequency.

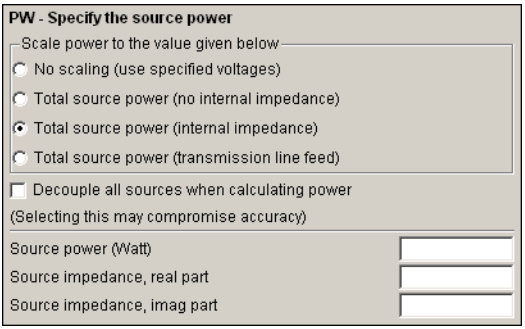
Note that the `*.str` file can be used for MoM, MLFMM, PO, and FEM, while the `*.mat` and `*.lud` files are only applicable when the standard MoM is used (and then also only for sequential in-core solutions).

Note that models built with PREFEKO on different computers may not be identical due to very small rounding differences of different CPUs. It is therefore advisable to run PREFEKO only on one computer when using this card, to ensure consistency in the `*.fek` files. The `*.fek` files can then be copied to another computer if required. (For example, a user may calculate and store the current distribution for a large model on a fast workstation and later load this to calculate different near fields on a small PC. To ensure that current solution is valid on the PC, the original `*.fek` file should be generated on the PC and copied to the workstation.)

If the PS card is used to specify that data should be read from a file, the PS card may occur only once in the input file. It is advisable to place it right after the EG card.

10.2.40 PW card

When defining the excitation of an antenna the source is normally specified as a complex voltage. The PW card allows the user to specify the radiated power or the source power instead (FEKO then just internally scales the whole solution to arrive at this desired power). In addition, it is possible to consider a mismatch between the antenna input impedance and the internal impedance of a voltage source or the characteristic impedance of a transmission line feed.


PW - Specify the source power
Scale power to the value given below
No scaling (use specified voltages)
Total source power (no internal impedance)
Total source power (internal impedance)
Total source power (transmission line feed)
Decouple all sources when calculating power
(Selecting this may compromise accuracy)
Source power (Watt)
Source impedance, real part
Source impedance, imag part

Parameters:

No scaling (use specified voltages): PW card is not activated, i.e. the specified value of the voltage source is used.

Total source power (no internal impedance): PW card is activated and all the currents are multiplied by a scaling factor such that the total source power (the sum of the power delivered by all the individual sources) is P_0 — the value specified in the *Source power* field. Mismatch is not considered.

Total source power (internal impedance): All voltage sources are assumed to have an input impedance Z_i as specified by the parameters *Source impedance, real part* and *Source impedance, imag part*. The currents are scaled such that the total power supplied by the voltage sources equals P_0 as discussed below. The mismatch losses in the source impedance reduce the antenna gain.

Total source power (transmission line feed): All the antennas are assumed to be fed by transmission lines with a complex characteristic impedance Z_L as specified by the parameters *Charact. impedance, real part* and *Charact. impedance, imag part*. If there is a mismatch between Z_L and the antenna input impedance Z_a, some of the incident power will be reflected back to the source.

Decouple all sources when calculating power: When this item is unchecked and multiple impressed sources (elementary dipoles A5/A6, impressed current elements AI/AV, etc.) are present, the mutual coupling of all these sources, as well as the coupling of the sources with other structures such as ground (BO card), UTD surfaces, or MoM elements are taken into account when determining the source power. This is also the default if the PW card is not present. When this item is checked, however, this mutual coupling is not considered. This is acceptable for sources which are relatively far away or when no accurate power values are required. (Since gain/directivity are based on power, they are then also possibly not very accurate.)

Source power: The total power P_0 in Watt, i.e. the total power supplied by all the voltage sources, or in the case of transmission lines, the total power of all forward travelling waves.

Details of the various possibilities with the use of the PW card are shown in figure 10-27.

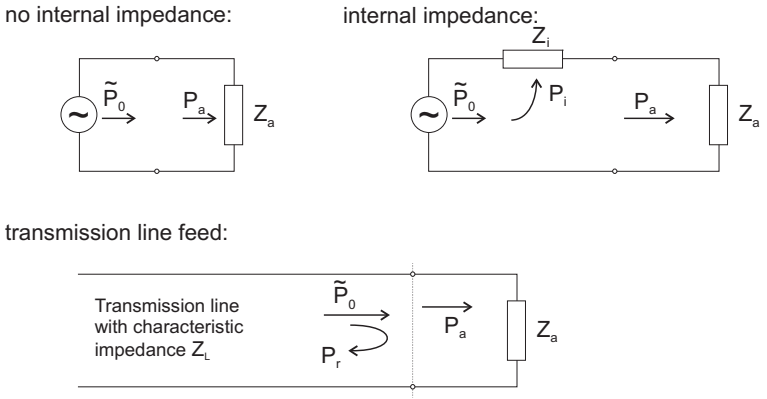


Figure 10-27: Possible applications of the PW card to determine the total power

The options *Total source power (internal impedance)* and *Total source power (transmission line feed)* are only allowed for voltage sources (the A1, A2, A3, A4, A7 and AE cards). For models containing other sources such as dipoles and impressed currents, the option *Total source power (no internal impedance)* should be used. For plane waves *No scaling (use specified voltages)* must be used.

The power equations for different cases are discussed below. Consider, in general, that there are N sources (such as in an array antenna) with open circuit voltages $U_{0,\nu}$ (before the scaling operation) where the parameter ν lies in the range $1 \ldots N$. At each source there is an antenna input impedance $Z_{a,\nu}$ (as calculated during the FEKO solution) to which power $P_{a,\nu}$ is transferred.

- *Total source power (no internal impedance)*

  Using this option all the source power is delivered to the respective antennas, i.e.

$$\tilde{P}_{0,\nu} = P_{a,\nu}$$

  as shown in figure 10-27. To ensure that the total power is P_0, the power must be scaled with the factor

$$s = \frac{P_0}{\sum_{\nu=1}^{N} \tilde{P}_{0,\nu}} = \frac{P_0}{\sum_{\nu=1}^{N} P_{a,\nu}}$$

  The currents on the structure are consequently scaled with the factor $\sqrt{s}$. There is no power loss.

- *Total source power (internal impedance)*

  When this option is used the internal impedance Z_i of the voltage source is considered as shown in figure 10-27. Since the same current flows through the internal source impedance and the antenna input impedance, the power dissipated in the impedance of the ν^{th} voltage source is given by the relation

$$P_{i,\nu} = P_{a,\nu} \frac{\operatorname{Re} Z_i}{\operatorname{Re} Z_{a,\nu}}$$

  and the scaling factor s (to scale the total power supplied by the sources to P_0) is

$$s = \frac{P_0}{\sum_{\nu=1}^{N} \tilde{P}_{0,\nu}} = \frac{P_0}{\sum_{\nu=1}^{N} (P_{a,\nu} + P_{i,\nu})} = \frac{P_0}{\sum_{\nu=1}^{N} P_{a,\nu} \left(1 + \frac{\operatorname{Re} Z_i}{\operatorname{Re} Z_{a,\nu}}\right)}$$

  The combined loss caused by the mismatched antennas

$$P_{\text{loss}} = s \sum_{\nu=1}^{N} P_{i,\nu} = s \sum_{\nu=1}^{N} P_{a,\nu} \frac{\operatorname{Re} Z_i}{\operatorname{Re} Z_{a,\nu}}$$

  reduces, for example, the antenna gain (but not the directivity).

- *Total source power (transmission line feed):*

  When this option is used each antenna (with input impedance $Z_{a,\nu}$) is considered to be excited by a transmission line with a complex characteristic impedance Z_L as shown in figure 10-27. For most practical applications the transmission line will be lossless, resulting in a real characteristic wave impedance. For this lossless case the reflection factor

$$\varrho_\nu = \frac{Z_{a,\nu} - Z_L}{Z_{a,\nu} + Z_L}$$

is taken into account when calculating the incident power at the feed point.

The total incident power is given by

$$\tilde{P}_{0,\nu} = \frac{P_{a,\nu}}{1 - |\varrho_\nu|^2}$$

and the reflected power by

$$P_{r,\nu} = |\varrho_\nu|^2 \, \tilde{P}_{0,\nu} = P_{a,\nu} \, \frac{|\varrho_\nu|^2}{1 - |\varrho_\nu|^2}$$

To ensure that the total incident power is P_0, the power is scaled with the factor

$$s = \frac{P_0}{\sum\limits_{\nu=1}^{N} \tilde{P}_{0,\nu}} = \frac{P_0}{\sum\limits_{\nu=1}^{N} \frac{P_{a,\nu}}{1 - |\varrho_\nu|^2}}$$

and the currents with the factor $\sqrt{s}$. As before the total reflected power

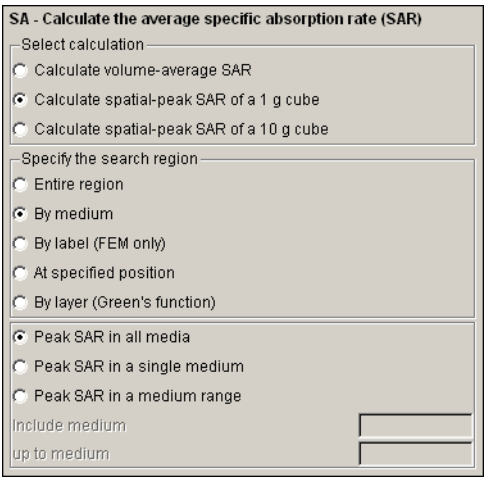
$$P_{\text{loss}} = s \sum_{\nu=1}^{N} P_{r,\nu} = s \sum_{\nu=1}^{N} P_{a,\nu} \, \frac{|\varrho_\nu|^2}{1 - |\varrho_\nu|^2}$$

reduces the gain of the antenna.

If we have a lossy transmission line, then forward and backward travelling waves can still be identified on such a transmission line, but a proper definition of power associated with such a wave is not possible, since due to the losses the power will constantly change along the length of the transmission line. It is also questionable whether using the PW card with a lossy transmission line makes any sense. But it has still been implemented, and then the forward travelling power P_0 is interpreted as the maximum available power at the end terminals of the transmission line. For a lossless transmission line this formulation is compatible with the above equations.

10.2.41 SA card

This card is used to control calculations of the specific absorption rate (SAR) in a dielectric medium.


SA - Calculate the average specific absorption rate (SAR)
Select calculation
Calculate volume-average SAR
Calculate spatial-peak SAR of a 1 g cube
Calculate spatial-peak SAR of a 10 g cube
Specify the search region
Entire region
By medium
By label (FEM only)
At specified position
By layer (Green's function)
Peak SAR in all media
Peak SAR in a single medium
Peak SAR in a medium range
Include medium
up to medium

Parameters:

Select Calculation: One of three SAR values which could be of interest should be selected in this group.

Specify the search region: This group can be used to control, either by medium number or by label number or by layer number (for the special Green's functions), which dielectric bodies are used for the specified calculation. It is also possible to specify a user defined position here for the spatial average SAR computations.

Average/Peak SAR in all media/labels/layers: Select this option if the selected SAR calculation should be done on a 'by label' or 'by medium' or 'by layer' basis. The whole body average SAR is also calculated. Selecting the volume by label is only valid for the FEMt analysis.

Average/Peak SAR in a single medium/label/layer: The selected SAR calculation is obtained for the medium/label specified in the *Include medium/label* dialog.

Average/Peak SAR in a medium/label/layer range: The selected SAR calculation is performed on the label range as specified below in the input fields for *Include medium/label* and *up to medium/label*.

Centre of SAR cube: For the spatial average SAR computations using a specified position, the X, Y and Z coordinates of the cube centre must be specified here.

The required SAR calculation is performed, and the result saved in the `*.out` file.

If the options *Calculate volume-average SAR* and *Entire region* are selected, the SAR, averaged over all media, is returned. If the options *Calculate volume-average SAR* and *By medium* are selected, the average SAR is calculated per medium and tabled in the `*.out` file. If a medium/label/layer range is specified, the SAR is averaged over the volume defined by the medium/label/layer range.

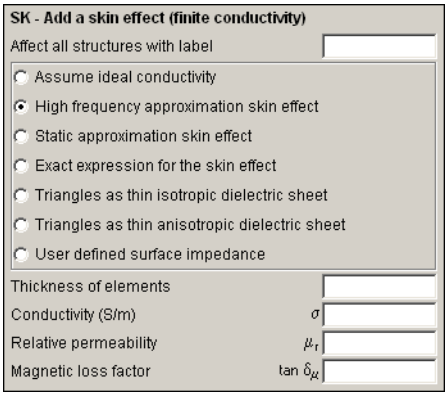
If a spatial peak SAR calculation is requested, then spatial-peak SAR is computed, averaged over a mass of either 1 g or 10 g of tissue in the shape of a cube. By default, the search for the spatial-peak SAR in the entire domain is returned, otherwise the spatial-peak SAR can be requested for regions in a specified medium/label/layer range, or also at a user specified position.

When a special spherical or multilayer planar Green's function is used in FEKO, then also spatial peak average SAR values can be computed (not volume average SAR). A selection is possible by a single layer number, a range of layer numbers, or by including the whole dielectric volume in the search.

10.2.42 SK card

This card is used to consider the Skin effect or ohmic losses or an arbitrary user defined impedance boundary condition on wire segments and surface elements. In addition, it can switch from metallic triangles to thin dielectric layers which may consist of multiple layers and which may be anisotropic. If no SK card has been defined, FEKO assumes ideal conductivity without any losses.

10.2.42.1 Skin effect and ohmic losses


SK - Add a skin effect (finite conductivity)
Affect all structures with label
Assume ideal conductivity
High frequency approximation skin effect
Static approximation skin effect
Exact expression for the skin effect
Triangles as thin isotropic dielectric sheet
Triangles as thin anisotropic dielectric sheet
User defined surface impedance
Thickness of elements
Conductivity (S/m) σ
Relative permeability μr
Magnetic loss factor tan δμ

Parameters:

Affect all structures with label: Affect all segments and triangles with this label.

Assume ideal conductivity: Ideal conductivity is assumed (also the default when there is no SK card for a given label). All other parameters are ignored.

High frequency approximation skin effect: High frequency approximation.

Static approximation skin effect: Static (ohmic losses) approximation.

Exact expression for the skin effect: Use the exact expression of the skin effect for wires and/or surfaces that is valid at arbitrary frequencies.

Thickness of elements: The thickness d of the surface elements in m (if an SF card is present, this is always scaled).

Conductivity: The conductivity σ in $\frac{1}{\Omega \mathrm{m}}$.

Relative permeability: The relative permeability μ_r of the material.

Magnetic loss factor: Magnetic loss tangent $\tan\delta_\mu$ (the complex permeability is then given by $\mu = \mu_0\mu_r\left(1 - j\tan\delta_\mu\right)$).

Having both triangles and segments with the label *Affect all structures with label* should be avoided. Separate labels and a distinct SK card for each label should be used. In addition all wires with the label *Affect all structures with label* must have the same radius. If this is not the case a unique label must be introduced for each radius.

It should be noted that the skin depth is given by $\delta_{skin} = \sqrt{\frac{2}{\omega\mu\sigma}}$, where the radial frequency $\omega = 2\pi f$ and the permeability $\mu = \mu_r\mu_0$.

- *Assume ideal conductivity:*
  No further parameters.
- *High frequency approximation skin effect:*
  The required parameters are μ_r, $\tan\delta_\mu$ and σ, and for surfaces also the thickness d. A good conductivity is required, i.e. $\sigma \gg \omega\varepsilon_0$.
  - For wires a further condition requires that $\delta_{skin} < \varrho$ where ϱ is the wire radius. The surface impedance is given by $Z'_s = \frac{1}{2\pi\varrho}\sqrt{\frac{j\omega\mu}{\sigma}}$.
  - For metallic surfaces the condition $\delta_{skin} < \frac{d}{2}$ must be met. The surface impedance is given by $Z_s = \frac{1}{2}\sqrt{\frac{j\omega\mu}{\sigma}}$.
- *Static approximation skin effect:*
  The required parameters are μ_r, $\tan\delta_\mu$ and σ, and for surfaces also the thickness d. A good conductivity is required, i.e. $\sigma \gg \omega\varepsilon_0$.
  - For wires a further condition requires that $\delta_{skin} > \varrho$ where ϱ is the wire radius. The surface impedance is given by $Z'_s = \frac{1}{\pi\varrho^2}\frac{1}{\sigma}$.
  - For metallic surfaces the condition $\delta_{skin} > \frac{d}{2}$ must be met. The surface impedance is given by $Z_s = \frac{1}{\sigma d}$.
- *Exact expression for the skin effect:*
  The required parameters are μ_r, $\tan\delta_\mu$ and σ, and for surfaces also the thickness d. A good conductivity is required, i.e. $\sigma \gg \omega\varepsilon_0$.
  - For wires with radius ϱ the surface impedance is given by

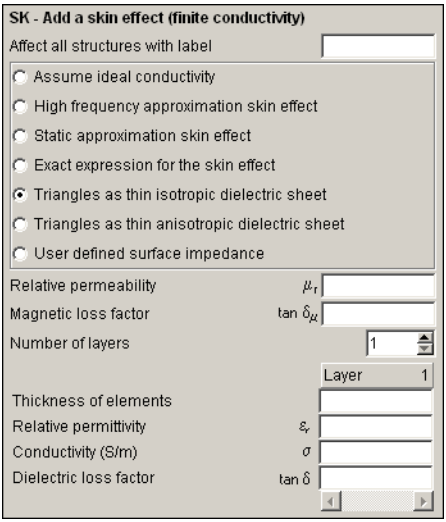
    $$Z'_s = \frac{1-j}{2\pi\varrho\sigma\delta_{skin}} \frac{J_0\left[(1-j)\frac{\varrho}{\delta_{skin}}\right]}{J_1\left[(1-j)\frac{\varrho}{\delta_{skin}}\right]}$$

    where J_0 and J_1 are Bessel functions.
  - For metallic surfaces the surface impedance is given by

    $$Z_s = \frac{1-j}{2\sigma\delta_{skin}} \frac{1}{\tan\left[(1-j)\frac{d}{2\delta_{skin}}\right]}$$

  Examples are given in `example_02` and `example_33` in the *Examples Guide*.

10.2.42.2 Triangles as a thin isotropic dielectric sheet



Parameters:

Relative permeability: The relative permeability μ_r which is the same for all layers.

Magnetic loss factor: Magnetic loss tangent $\tan\delta_\mu$ — the complex permeability is then $\mu = \mu_0\mu_r\,(1 - j\tan\delta_\mu)$ — which is the same for all layers.

Number of layers: The number of layers. If this field is empty or 0, it defaults to 1.

Thickness of elements: The thickness d of the current layer in m (if an SF card is present, this is always scaled).

Relative dielectric constant: The relative dielectric constant ε_r of the current layer.

Conductivity: The conductivity σ in $\frac{1}{\Omega\mathrm{m}}$ of the current layer.

Electric loss factor: The loss tangent $\tan\delta$ ($\tan\delta = \frac{\sigma}{\omega\varepsilon_0\varepsilon_r}$) of the current layer.

This option only makes sense for triangular surfaces, not for wires. The required parameters are d, μ_r, $\tan\delta_\mu$ and ε_r as well as σ or $\tan\delta$, such that $\mu = \mu_r\mu_0$ and the complex dielectric constant $\varepsilon = \varepsilon_r\varepsilon_0\,(1 - j\tan\delta) - j\frac{\sigma}{\omega}$. Normally either σ or $\tan\delta$ is entered as zero, but it is possible to specify both (for example, to approximate a specific frequency response). FEKO will give a warning which may be ignored.

The triangles with the label *Affect all structures with label* exist in a certain environment ε_e, μ_e, which is usually specified with the EG card. It is also possible to place the triangles

within a dielectric body — in this case the environment is specified by the parameters of the DI card. An additional condition is that the triangles should be geometrically thin, i.e. d must be small relative to the lateral dimensions. The mesh size is determined by the wavelength in the environment (i.e. in the medium ε_e, μ_e). Thus the layers must be thin relative to the wavelength in the environment.

When used with the MoM, the use of *Triangles as thin isotropic dielectric sheet* requires that $\mu = \mu_e$ and $\varepsilon \neq \varepsilon_e$. For a single layer, the card consists of only one line. The surface impedance, as used by FEKO, is then

$$Z_s = \frac{\beta}{2j\omega\left(\varepsilon - \varepsilon_e\right)\sin(\beta\frac{d}{2})}$$

where $\beta = \omega\sqrt{\varepsilon\mu}$ is the propagation constant. An example is given in `example_32` (*Examples Guide*). For multiple layers the card requires one line per layer with the parameters of the first layer on the same line as the card name. The approximate surface impedances of the different layers are added to determine the effective surface impedance.

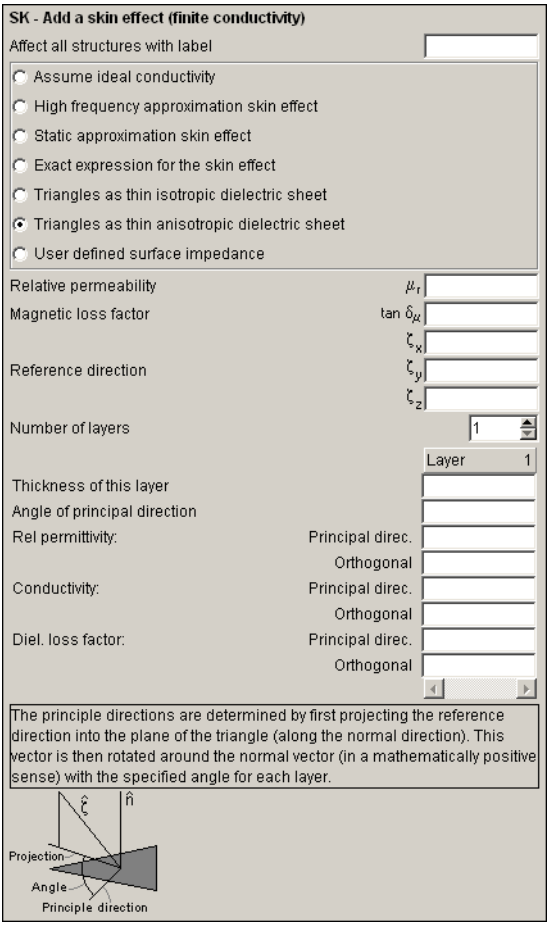
When used with PO, it is not required that $\mu = \mu_e$ or $\varepsilon \neq \varepsilon_e$. In this case the order of the layers is also significant. The layer on the side that the triangle normal vector points to, is specified in the first line with the remaining layers following in sequence.

10.2.42.3 Triangles as a thin anisotropic dielectric sheet

This option is very similar to *Triangles as thin isotropic dielectric sheet,* but the layers are anisotropic. The principle direction in each layer is defined by the angle α (*Angle of principle direction* field) relative to the projection of the vector ζ (*Reference direction* field) onto the plane of triangle. Here α is measured in the mathematically positive sense with respect to the normal vector of the triangle. (POSTFEKO can be used to display the fibre direction and visually check that the input file is correct.) In this case the card line is followed by an additional line for each layer. The medium properties in the *principle direction* is different from those in the *orthogonal direction* which lies in the plane of the triangle and orthogonal to the principle direction.

Parameters:

Relative permeability:	The relative permeability μ_r which is the same for all layers.
Magnetic loss factor:	Magnetic loss tangent $\tan\delta_\mu$ — the complex permeability is then $\mu = \mu_0\mu_r\left(1 - j\tan\delta_\mu\right)$ — which is the same for all layers.
Reference direction:	The x, y and z components of the vector ζ (used to define the principle direction, see *Angle of principal direction*).
Number of layers:	The number of layers. If this field is empty or 0, it defaults to 1.
Thickness of this layer:	The thickness d of this layer in m (if an SF card is present, this is always scaled).


SK - Add a skin effect (finite conductivity)
Affect all structures with label
Assume ideal conductivity
High frequency approximation skin effect
Static approximation skin effect
Exact expression for the skin effect
Triangles as thin isotropic dielectric sheet
Triangles as thin anisotropic dielectric sheet
User defined surface impedance
Relative permeability
Magnetic loss factor
Reference direction
Number of layers
Layer 1
Thickness of this layer
Angle of principal direction
Rel permittivity:
Principal direc.
Orthogonal
Conductivity:
Principal direc.
Orthogonal
Diel. loss factor:
Principal direc.
Orthogonal
The principle directions are determined by first projecting the reference direction into the plane of the triangle (along the normal direction). This vector is then rotated around the normal vector (in a mathematically positive sense) with the specified angle for each layer.
Projection
Angle
Principle direction

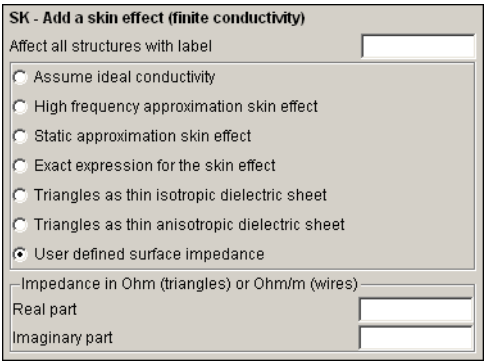
Angle of principle direction: The angle α_i (in degrees) from which the principle direction is obtained, see note above.

Rel permittivity: The relative dielectric constant ε_r of this layer in the *Principle* and *Orthogonal* directions respectively.

Conductivity: The conductivity σ in $(\frac{1}{\Omega \mathrm{m}})$ of this layer in the *Principle* and *Orthogonal* directions respectively.

Loss factor: The loss tangent $\tan\delta$ ($\tan\delta = \frac{\sigma}{\omega\varepsilon_0\varepsilon_r}$) of this layer in the *Principle* and *Orthogonal* directions respectively.

10.2.42.4 User defined surface impedance


SK - Add a skin effect (finite conductivity)
Affect all structures with label
Assume ideal conductivity
High frequency approximation skin effect
Static approximation skin effect
Exact expression for the skin effect
Triangles as thin isotropic dielectric sheet
Triangles as thin anisotropic dielectric sheet
User defined surface impedance
Impedance in Ohm (triangles) or Ohm/m (wires)
Real part
Imaginary part

With this option an arbitrary user defined complex surface impedance $Z_{\square}$ can be used in FEKO. It must be noted that the impedance boundary condition for the MoM (also then for MLFMM etc.) has certain limitations regarding the range of validity. FEKO uses whatever the user has specified as surface impedance, and it is the responsibility of the user to ensure that the application of an IBC is still warranted for the specific configuration (dependent on the impedance value as such, but also frequency, radius of curvature of the structure etc.).

For surfaces the unit of $Z_{\square}$ is Ω. For wire structures, the value used by FEKO is in units of $\frac{\Omega}{\mathrm{m}}$ and results from the surface impedance expression by dividing it by $\pi\varrho$ where ϱ represents the wire radius.

Parameters:

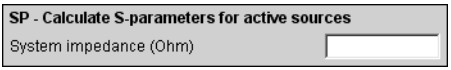
Real part:	The real part of the surface impedance $Z_{\square}$ in Ω (for triangles) or in $\frac{\Omega}{\mathrm{m}}$ for wires.
Imaginary part:	The imaginary part of the surface impedance $Z_{\square}$ in Ω (for triangles) or in $\frac{\Omega}{\mathrm{m}}$ for wires.

For instance, to model a solid dielectric object with relative permittivity ε_r and with conductivity σ at a specific angular frequency $\omega = 2\,\pi\,f$, one could use the surface impedance expression

$$Z_{\square} = \sqrt{\frac{\mu_0}{\epsilon_0\varepsilon_r - j\frac{\sigma}{\omega}}} \quad .$$

10.2.43 SP card

This card is used to calculate the S-parameters for the active sources.



Parameters:

System impedance: The reference impedance in Ohm. This is used for all sources for which no impedance value is specified when defining the source. If this field is empty, it defaults to 50 Ω. Note that for waveguide sources (AW card) S-parameters are always related to the corresponding waveguide impedance.

The N ports must be defined before using the SP card. They are identified simply by defining excitation cards. Currently only A1, A2, A3, A4, AE, and AW sources are supported. A1, A2 and A3 sources must be selected by label (not with position), and unique labels must be used (i.e. no other segments or triangles may have a label which is used for a port).

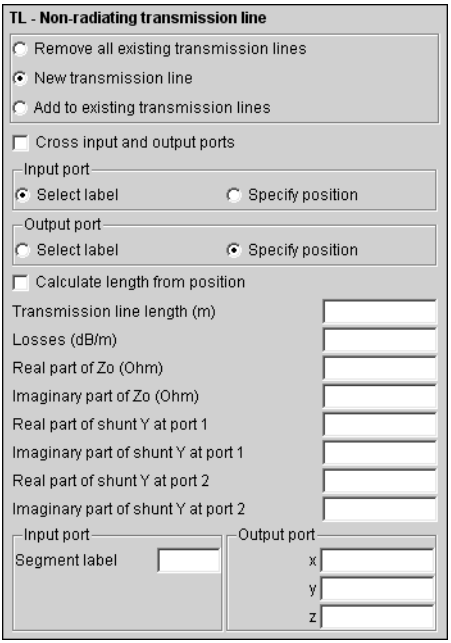
If the amplitude of any port is set to zero, it will be used as a receive port (or sink) but not as a source. For example, if only S_{21} and S_{11} are required for a two port network, one may set the amplitude of the source defining port 2 to exactly zero. Then S_{12} and S_{22} are not calculated — in some cases this may save considerable computation time.

The load impedance for each of the port sources can be specified at the source itself. If no such impedance was specified, the *System impedance (Ohm)* value specified with the SP card will be used (if this value is not specified it defaults to 50 Ω). This load impedance will be added automatically to each port.The only exception here are waveguide ports (AW card), where S-parameters are related directly to the corresponding waveguide impedance.

It must be noted that except for waveguide ports the SP card adds load impedances to all the ports. (For A1, A2 and A3 sources it uses LZ type loads if no transmission line is connected to these segments, otherwise the shunt admittance of the transmission line is set directly; for A4 sources it uses L4 type loads and for AE sources it uses LE type loads.) If any similar loads were applied to the source position before the SP card these loads will be overwritten. Also when execution continues after processing the SP card these loads will still be present. The original amplitudes of the excitations will, however, be restored.

10.2.44 TL card

This card is used to connect a non-radiating transmission line between two segments.



Parameters:

Remove all existing transmission lines: If checked, all previously defined transmission lines are deleted. All the other input parameters are ignored.

New transmission line: Defines a new transmission line, all previously defined transmission lines are replaced.

Add to existing transmission lines: An additional transmission line is defined.

Cross input and output ports: The positive port voltage is in the direction of the segment that it is connected to (from the start to the end point of the segment). Thus the input and output ports of the transmission line have unique orientations. If this item is checked the transmission line connecting the ports is crossed.

Input port: If *Select label* is selected in the upper group, the input port (the segment which represents the start of the transmission line) is defined by specifying the label of the segment in the *Segment label* field in the lower group. If more than one segment with the same label exists, then the last segment with this label is used.

If *Specify position* is selected in the upper group, the input port is defined by specifying the Cartesian coordinates of the centre of the segment.

Output port: Same as for *Input port,* but applies to the end of the transmission line.

Calculate length from position: If checked, FEKO determines the length based on the geometrical distance between the start and end points.

Transmission line length: The length of the transmission line in metres. This value is scaled with the scaling factor of the SF card.

Losses: Losses of the transmission line in dB/m. Note that since the propagation constant is taken as the propagation constant of the medium in which the start and end segments are located, the attenuation specified by this parameter is added to any losses of this medium. This factor is not affected by scaling specified with the SF card, i.e. if a scaling factor which reduces the length of the transmission line is added, the total loss through the line will be less. (The length is now less and the loss per distance remained the same.)

Real part of Zo: Real part of the characteristic impedance of the transmission line in Ohm.

Imaginary part of Zo: Imaginary part of the characteristic impedance of the transmission line in Ohm. Note that the characteristic impedance only defines the ratio between the voltage and current of the two waves propagating along the line. It does not specify any losses.

Real part of shunt Y at port 1: Real part of the shunt admittance at the input port in Siemens. (This admittance is across the port, connecting the two wires of the transmission line.)

Imaginary part of shunt Y at port 1: Imaginary part of the shunt admittance at the input port in Siemens.

Real part of shunt Y at port 2: Real part of the shunt admittance at the output port in Siemens. (This admittance is across the port, connecting the two wires of the transmission line.)

Imaginary part of shunt Y at port 2: Imaginary part of the shunt admittance at the output port in Siemens.

An arbitrary number of transmission lines can be used, also one wire segment could be for instance the end of one transmission line and the start of another. All transmission lines ending at one wire are connected in parallel.

Any load impedance defined over the transmission line port segments with the LZ, LS, LP, LD, CO or SK cards are placed in series with the port, parallel admittances can be defined directly at the TL card.

If a voltage source of type A1 or A3 is applied at one of the port segments, then this voltage source is assumed to be across the port (i.e. feeding the transmission line directly with an impressed voltage). Any other sources are in series with the port.

FEKO automatically determines the type of the transmission line network:

1 no impressed voltage source at both ports
2 impressed voltage source at the input port
3 impressed voltage source at the output port
4 impressed voltage source at both ports

The wire segments for the two ports should be located in the same medium, so that the propagation constant of this medium can be taken for the transmission line. If the segments are in different media, then the medium of the segment at the input port is used. Note that the propagation constant and thus also the propagation loss of the transmission line is the same as that of the medium surrounding input port unless an additional loss factor is specified in the *Losses* field. If this is free space the transmission line will be lossless. For transmission lines with a propagation constant that is higher than that of the surrounding medium, such as coaxial cables filled with dielectric material, one needs to reduce the length of the transmission line.

Losses in the transmission line network (due to the shunt admittances or transmission line losses directly) are taken into account and will for instance reduce antenna efficiency or gain.

The TL card is used in `example_39` (*Examples Guide*) to create a log periodic antenna.

11 The optimiser OPTFEKO

Optimising is the process of changing parameters in a model to achieve a specific response. These parameters are usually associated with geometric entities, material properties or excitations/loadings. For example, the maximum gain of property of a horn antenna can be optimised by varying the size of the horn opening.

As discussed in section 7.3 the program PREFEKO can handle symbolic variables. All parameters that must be optimised must use symbolic variables. OPTFEKO changes the value of these variables, in a predictable way, until the desired behaviour is obtained.

Apart from defining the optimisation parameters, the aim (or goal) of the optimisation must be specified. An aim function is used to determine how close the current solution comes to the desired goal. The optimisation routine always finds the minimum, so the aim function should be defined such that its minimum is the optimum solution. OPTFEKO also makes provision for multiple aim functions to be used in one optimisation.

Unfortunately, no optimisation algorithm can guarantee an optimum solution by specifying only the parameters and the aim. FEKO supports a variety of optimisation functions. Each one is discussed along with its own parameters in this section.

11.1 Preparing input for OPTFEKO

OPTFEKO requires two files for successful execution. The first is a `*.pre` file, and the second is an `*.opt` file. The contents of both of these files is prepared using the preprocessor EDITFEKO, see Chapter 5.

11.1.1 More on the *.pre file

The first required file is the normal `*.pre` input file for PREFEKO, where the symbolic variables are used to define (at least) the part of the model that needs to be optimised. During optimisation these variables are varied by OPTFEKO. This is done by repeatedly generating new `*.pre` files, in which the optimisation parameters are assigned to the symbolic variables. The user usually needs to have 'default' values for these variables, that can be used when running PREFEKO to view the geometry, prior to running OPTFEKO. This must be done by using the DEFINED function. An example is given in section 11.4).

11.1.2 More on the *.opt input file

The optimisation method, its parameters and the aim of the optimisation are stored in a second file with the same filename as the `*.pre` file and an `*.opt` extension. Since the EDITFEKO GUI is used to prepare this file, only a general understanding of the file structure is necessary.

This file consists of three or four sections:

- Assignment of optimisation parameters, minimum and maximum values
- Optional assignment of the penalty function
- Optimisation parameters (e.g. step size, final value)
- One or more aim functions with the required parameters

In the `*.opt` file blank and/or comment lines (starting with `**`) are allowed between the sections, but they are not allowed to appear in the sections themselves. The parameters are space delimited, i.e. they are separated by spaces and can be placed in any column, but they must be entered in the correct order.

The keywords used in the `*.opt` file exist in both German and English (for example, `RASTERSUCHE` and `GRID_SEARCH`). OPTFEKO will recognise keywords in either language, independent of the language selected by the environment variable `FEKOLANG`.

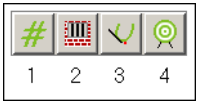
11.2 Using EDITFEKO to create a *.opt file

The EDITFEKO GUI is used to control the creation of the .opt file. To learn more about launching and using EDITFEKO, consult Chapter 5. The format and creation of the `*.pre` file is discussed in this and other chapters. All the functionality normally available in PREFEKO files is available when optimising.

Only the application of EDITFEKO to editing `*.opt` files is discussed in this chapter. In order to use EDITFEKO to edit this file, simply open the file, if it exists, or select the creation of a new `*.opt` file. This will bring up a blank window on the right and a panel on the left with a single *Edit* button. Press this button to open the dialog panel for the optimisation parameters.

Note that this edit mode has a lot in common with editing `*.pre` files. If an existing `*.opt` file is open in the editor, pressing F1 in a section of the file will open the editing panels for that section.

11.2.1 The optimising toolbar



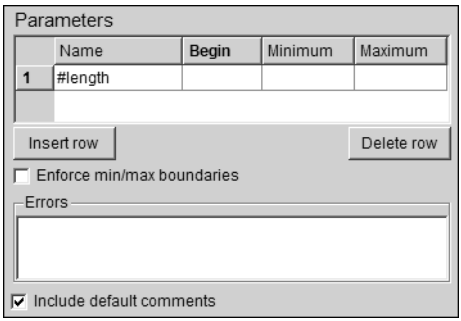
At the top of the optimisation editing panels, the optimising toolbar is shown. This toolbar is used to switch between the panels that control the four aspects of optimisation.

The functions of the buttons are:

1. Show the parameter control panel. This panel is used to specify which parameters must be optimised.
2. Show the penalty function control panel. This panel is used to specify penalty functions for variables that exceed given boundaries.
3. Show the optimisation function control panel. This panel is used to choose the optimisation method that OPTFEKO must use, as well as its' parameters.
4. Show the aim function control panel. This panel is used to specify the goal(s) of the optimisation function.

11.2.2 Definition of optimisation parameters

The parameters to optimise are controlled using this panel. These parameters correspond to variables in the `*.pre` file.



Parameters: The optimisation parameters are defined in tabular form in this field. Each optimisation parameter has a name (a symbolic variable used, but not numerically defined, in the `*.pre` file). For each parameter a begin value as well as a minimum and maximum value have to be given. The name must include the # character.

Insert row: This button is used to add a row to the table.

Delete row: This button is used to remove a row from the table.

Enforce min/max boundaries: Check this box to enforce sharp boundaries on the parameter. See the discussion below.

Errors: Any errors in the `*.opt` file that were noticed when the file was parsed are shown in this dialog.

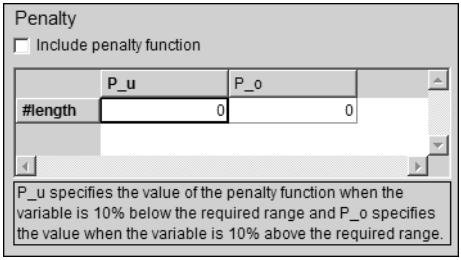
Include default comments: This box should be checked if the default descriptive comments should be added to the file. This checkbox appears at the bottom of all OPTFEKO panels.

Normally the minimum and maximum values of the optimisation parameters as specified in the `*.opt` file are only used for the normalisation of the parameter space. In order to ensure that the parameters stay within certain limits, a penalty function (see section 11.2.3) can be added. This will result in a smooth aim function for the optimisation.

In some circumstances, however, sharp boundaries must be enforced to avoid invalid geometries (for example, the distance between points cannot become negative). If the checkbox `Enforce min/max boundaries` is checked, the keyword `ENFORCE_MIN_MAX` or `ERZWINGE_MIN_MAX` is added to the `*.opt` file. The provided boundaries are then strictly enforced when OPTFEKO creates a `*.pre` file for FEKO to ensure its validity. This means that FEKO will produce meaningful results for any parameter value. OPTFEKO assumes that the obtained result is for the ORIGINAL parameter set (before enforcing the boundaries). This means that a penalty function should still be added in order to force the optimisation algorithms to move back into the valid parameter range.

11.2.3 Definition of the penalty function

Penalty functions are used to ensure that the optimal result is obtained for realistic values of the parameters. If the optimisation function calculates the response for parameter values that are outside the realistic range, then a penalty is added to the aim function, which effectively forces the optimisation function to consider this combination of values as non-optimal.


Penalty
Include penalty function
P_u
P_o
#length
0
0
P_u specifies the value of the penalty function when the variable is 10% below the required range and P_o specifies the value when the variable is 10% above the required range.

Include penalty function: This check box must be checked if penalty functions should be used for the optimisation.

Penalty function parameters: The left hand side of this table will contain the names of all the parameters. The two columns contain the over- and under-estimated penalty factors as described below.

To ensure that the optimisation parameter x stays within the bounds $x_{\min}$ and $x_{\max}$, a penalty function can be added to the aim function. The penalty function P is defined by the equation

$$P = \begin{cases} 10 \cdot P_u \cdot \frac{x_{\min} - x}{x_{\max} - x_{\min}} & \text{for} \quad x < x_{\min} \\ 0 & \text{for} \quad x_{\min} \leq x \leq x_{\max} \\ 10 \cdot P_o \cdot \frac{x - x_{\max}}{x_{\max} - x_{\min}} & \text{for} \quad x > x_{\max} \end{cases}$$

The two parameters P_u and P_o define the value of the penalty function when overestimating the allowable range by 10%. When the penalty function is added to the aim function, it must be ensured that they both are of the same dimension.

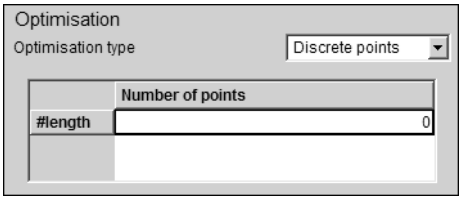
The penalty function is defined with the keyword `PENALTYFUNKTION` or `PENALTY_FUNCTION`. In the rows that follow, the optimisation parameters P_u and P_o are specified. If no penalty functions are to be assigned at all, then the whole section can be left out.

11.2.4 Definition of the optimisation process

During optimisation, new values for the parameters are chosen by a function known as the optimisation function. This panel is used to choose the optimisation function and its parameters. Each function has different parameters that must be set, and therefore each one is discussed separately below.

Discrete points

This method is strictly speaking not an optimisation method. The optimisation parameters are linearly varied between their minimum and the maximum values. This can be useful to investigate the variable space before beginning the optimisation. Due to the required computational time, this is not recommended for more than two variables. The value of the aim function is calculated at the specified discrete points. No penalty function may be specified in this case.



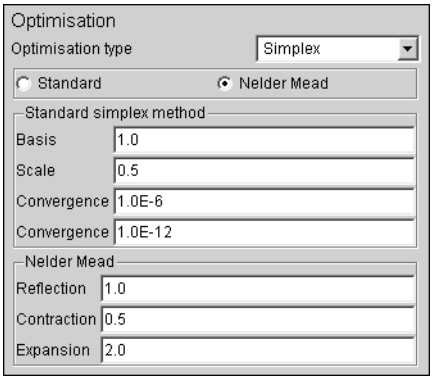
To use this option, the only parameters that are required are the number of points at which each optimisation parameter must be sampled. This value is entered into the table on the panel. The panel can also be used to specify that the default descriptive comments should be added to the `*.opt` file.

The keyword `GRID_SEARCH` or `RASTERSUCHE` is added to the `*.opt` file when choosing this function. The number of discrete points required for each of the optimisation parameters are specified in the line following the keyword.

If this function was applied to a problem with two optimisation parameters, where 8 samples of the first parameter and 9 samples of the second parameter are requested, FEKO will be executed 72 times. This example is given to highlight the computational cost of using this function for a large number of optimisation parameters.

Simplex method

The simplex method of optimisation finds the minimum of the aim function, by stepping through the optimisation space, in a direction determined by the last $n+1$ samples, where n is the number of optimisation parameters.



Standard: This option is selected to use the standard simplex method, where termination of the optimisation is determined from the size of the base of the simplex. The base of the simplex reduces in size when the simplex starts to rotate about one node. The standard simplex method requires at least two optimisation variables, if there is only one use the Nelder Mead variant.

Nelder Mead: This option specifies that the Nelder Mead variation of the simplex method should be used. This method allows for the simplex to reflect, contract and expand. The scaling factor for each action can be specified separately.

Basis: The initial size of the base of the simplex.

Scale: This factor specifies the factor by which to scale the base, if the simplex starts to rotate about a point.

Base size termination: The solution will be considered to be converged when the size of the base of the simplex is smaller than this value.

Deviation termination: If the standard deviation of the aim function at the nodes of the simplex is less than this value, the optimisation will terminate.

Reflection: This value determines the scale factor that will be applied to every reflection operation.

Contraction: This value determines the scale factor that will be applied to each contraction operation.

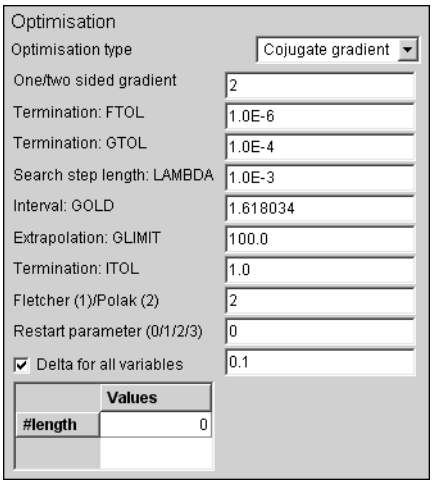
Expansion: This value determines the scale factor that will be applied to each expansion operation.

The keyword `SIMPLEX_METHOD` or `SIMPLEXVERFAHREN` is added to the `*.opt` file to use this optimisation function.

Conjugate gradient method

This optimisation method calculates the gradient of the aim function at each evaluation point. Based on the gradient at this point, as well as the gradient at previous points, a new search direction is chosen. A line search finds the minimum of the aim function in this direction. The next point for the optimisation function is at this point.

The new search direction can be calculated using one of two methods in FEKO. The first is the Fletcher-Reeves algorithm and the second is the variation by Polak-Ribiere. In general, the Polak-Ribiere variation is preferred, since it allows restarting.



One/two sided gradient: FEKO can calculate either the one or two sided gradient. The use of two sided gradients requires more FEKO runs per optimisation point, but generally provides better starting points for subsequent runs.

Termination: FTOL: This termination parameter sets the value by which two subsequent runs must differ in order for the optimisation to continue. If the difference between the aim functions starts to become very small, then a optimum has been reached.

Termination: GTOL: The maximum value of all the gradients at a point is calculated. If this value is less than this parameter, the optimisation is terminated.

Search step length: LAMBDA: This parameter is the search step size for the line search. It is advisable to keep this parameter small, since large values can result in a FEKO run that is very far from the current point.

Interval: GOLD: This parameter is important in determining the minimum and should be left at its default value of 1.618034.

Extrapolation: GLIMIT: This parameter is important in determining the minimum and should be left at its default value of 100.

Termination: ITOL: ITOL is the criterion for the termination of the one dimensional line search during reduction of the interval. This termination occurs if the value for 'lambda' deviates sufficiently little from the value which gives the arithmetic mean for the interval containing the minimum. Here 'lambda' is the factor with which to proceed in the given direction in order to minimise the aim function. The product of ITOL and the present value of 'lambda' yields the factor TOL1 which, among other characteristics, can lead to the termination of the interval reduction. Due to the implemented procedure, the present value for 'lambda' can, in such a case, be already a few orders of magnitude smaller than the specified value of LAMBDA. In case this value was chosen very small (for example 1.0E-3), the value used for ITOL should be increased accordingly (for example 1.0) so that the number of interval reductions executed per line search stays within limits and that the optimisation is not terminated prematurely.

Fletcher(1)/Polack(2): This selects the desired procedure to determine the new direction for the line search from the current gradient. A '1' indicates that the new direction will be determined as by the Fletcher-Reeves algorithm, whereas a '2' refers to the variation by Polak-Ribiere. In general the Polak-Ribiere formulation is preferred as it results in less steps to minimise the aim function and, to a certain extent, allows a restart.

Restart parameter (0/1/2/3): This parameter sets the cyclical restart of the (complete) Fletcher-Reeves method from the present point. It also determines, whether, and if so, after how many iterations (equal to the number of line searches plus one, or the number of gradient calculations plus two), the direction is periodically switched to

the steepest descent. In the case of non-harmonic functions this leads to better convergence, while the quadratic convergence of harmonic functions is preserved provided that the restarts do not occur more often than every N iterations (N being the number of variables to be optimised). The following values are possible:

- 0: No new starts
- 1: Restart after every (N) iterations
- 2: Restart after every (N+1) iterations
- 3: Restart after every (N+2) iterations.

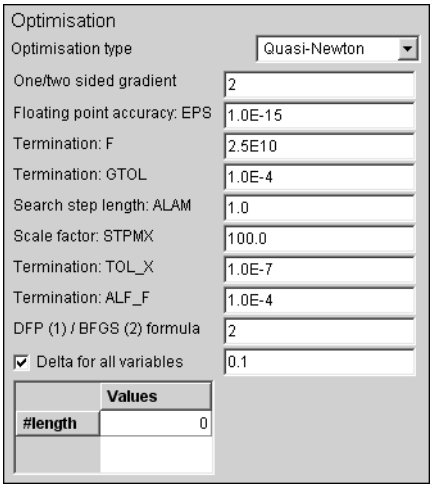
Restarts after (N+1) have proven to be reliable. Restarts are registered in the log and/or debug files in which the total number of line searches and gradient calculations is given sequentially. The variant of Polak-Ribiere does not allow any explicit restarts and this parameter must be initialised to 0.

Delta for all variables: DELTA establishes the 'dx' for the calculation of the gradient (single- or double-sided) and allows a separate 'dx' for every variable according to the assumed sensitivity of the aim function for this variable. The check box on the left can be checked if OPTFEKO should use the same value of Delta for all variables. If a different value should be used, the values should be entered in the table below.

The keyword for this optimisation method is `KONJUGIERTE_GRADIENTEN_VERFAHREN` or `CONJUGATE_GRADIENT_METHOD`.

Quasi-Newton method

This optimisation function uses the Quasi-Newton method by Davidon-Fletcher-Powell (DFP) or by Broyden-Fletcher-Goldfarb-Shanno (BFGS). The gradient is determined at a starting point in the variable space. The direction for the line search is then determined from the starting point and product of the negative gradient at this point and the unit matrix. The length of the search step is established in a one dimensional minimisation process which is distinct from the Conjugated Gradient method, both in the nature of the process and its accuracy. At this newly established point in the variable space, a new gradient is calculated. From the new and previous gradients as well as the new and previous points, in accordance with the DFP or BFGS formula for the metric, an approximation of the inverse Hesse matrix is now applied instead of the unit matrix. The product of this current approximation of the inverse Hesse matrix and the new negative gradient gives the direction for the next iteration. The method is terminated if the new gradient is sufficiently small or when the current point is within a pre-defined distance from the previous point.



One/two sided gradient: FEKO can calculate either the one or two sided gradient. Two sided gradients requires more FEKO runs per optimisation point, but generally provides better starting points for subsequent runs.

Floating point accuracy EPS: EPS is the accuracy to use for floating point calculations on the computer. Once adjusted, the value can be maintained. For real variables the value is 3.0E-8, whereas for double variables it is 1.0E-15. Here double variables have been used.

Termination: F: F sets the factor with which the tolerance of the variable vector x should be larger than the accuracy EPS. The product F*EPS is the upper limit for the maximum difference in any component of the last two aim vectors. It serves as a terminating condition for the whole process. For maximum accuracy, experience suggests using a value of four. If less accuracy is needed, reaching the terminating condition can be accelerated by choosing F very large (for example 2.5E10). The further apart the two last aim vectors, the 'harder' the condition to abort the procedure due to convergence of the variable vector.

Termination: GTOL: GTOL represents the size of the maximum of all the components of the gradient at the current point. The iteration terminates if all the components of the gradient are smaller than this value.

Search step length: ALAM: ALAM designates the dynamic length of the search step in the line search - similar to the increment factor LAMBDA in the line search used in Brent's method (see the Conjugate Gradient method). A Newton step is executed at the start of every line search. In the immediate vicinity of a minimum, this leads to quadratic convergence when ALAM is set to 1.0. The factor for

ALAM should only be changed in exceptional cases (variational procedures). With further search steps within a line search ALAM is determined first by quadratic, then by cubic approximation.

Scale factor: STPMX: The factor for STPMX is used to scale the aim function should it become necessary to limit the maximum step width during a line search. This avoids evaluating the aim function at points where it could overflow or where it has not been defined. At the start of the optimisation process, an internal factor is calculated from the STPMX variable and the components of the start vector. This is then used to scale all further calculations on aim vector in a similar manner. If at the beginning of a line search, the square root of the sum of the squared components (the absolute value of the aim vector) is larger than this factor, all components of the target vector are uniformly multiplied with a value < 1 (with reference to the actual value of this vector), thus reducing the length of the aim vector. The default value for STPMX is 100.

Termination: TOL_X: TOL_X is also a termination factor (in addition to the convergence is above the target vector) for the line search. It is related to the maximum component of the quotient of the current direction and maximum component of the previous aim vector. This expression (alamin) vector. This expression (alamin) is the lower bound of the step width factor 'lambda' (alam) during the examination of the value of the aim function in a direction. The line search is terminated when alam $<$ alamin. The standard value for TOL_X is 1.0E-7.

Termination: ALF_F: ALF_F presents another method to terminate the line search. This happens when there is a sufficient decrease of the current value of aim function in comparison with the previous one. "Sufficient" here refers to the fact that the current value must not only be absolutely smaller, but must be smaller than the previous value less the factor ALF*Starting_Gradient where Starting_Gradient refers to the gradient at the previous point. The standard value for ALF = 1.0E-4.

DFP (1)/ BFGS (2) formula: This parameter is used to choose the desired process to determine the direction for the new line search from the gradient calculation; the parameter '1' indicates that the new direction is determined using the approximation of the inverse Hesse matrix according to the DFP formula, while the parameter '2' indicates the use of the BFGS formula to accomplish the same task. The two methods to determine the new direction differ only slightly in their effect on the total process, for example in rounding and convergence errors. It has, however, been shown that the BFGS variant is sensitive to non-optimal line searches.

Delta for all variables: DELTA establishes the 'dx' for the calculation of the gradient (single- or double-sided) and allows a separate 'dx' for every variable according to the assumed sensitivity of the aim function for this variable. The check box on the left can be checked if OPTFEKO should use the same value of Delta for all variables. If a different value should be used, the values should be entered in the table below.

The keyword used for this method is `QUASI_NEWTON_VERFAHREN` or `QUASI_NEWTON_METHOD`.

11.2.5 Defining the aim function

A number of different aim functions are available in OPTFEKO. These aim functions represent typical electromagnetic parameters that need to be optimised, such as radiation patterns and input impedance. The aim function specifies the desired response from the model to be optimised.

Data from different blocks (due to frequency change etc.) can be combined within an aim function in two different ways. We differentiate between the local aim function, and the global aim function. The local aim function f_i is defined per solution block. The global aim function Z is calculated from a combination of the local aim functions. In some cases, the global aim function is defined directly, since it does not make sense for these functions to be calculated on a per block basis. In general, the global aim function can either aim to reduce the average of the local aim functions or to minimise the maximum of the local aim functions. If an average local aim function should be minimised, the global aim function Z is defined by

$$Z = \frac{1}{N_B} \cdot \sum_{i=1}^{N_B} f_i,$$

If the maximum local aim function should be minimised (minimax principle), the global aim function is defined by

$$Z = \max_{i=1\ldots N_B}(f_i)$$

In this section the errors, f_i, for each of the target quantities is defined.

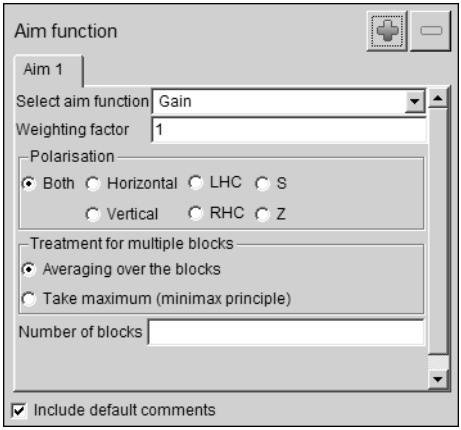
When doing optimisation, quite often there is not only one optimisation target. For instance one might want to optimise the antenna gain but at the same time get the impedance right. Or one might want to optimise an antenna pattern, and use different shape functions for the horizontal and vertical cut-planes. Each global aim function specifies one type of observable to be optimised. Any number of these aim functions can be combined linearly to optimise more than one observable at a time. In this case, each of these aim functions get an additional weighting factor associated with it. This weighting factor specifies the relative contribution of each individual aim function to the total aim function. If the weighting factor for an aim function is not specified, the default value of 1 is used.

Where relevant the following terms are used in this section:

- N_B is the number of blocks in the output file that must be used for the optimisation.
- N_S is the number of samples per block in the output file.
- f_i is, as above, the aim function as defined per block.

Gain

With this aim function a maximisation of the gain/directivity of an antenna in one or more directions can be done. The optimisation can be done over a broad band by examining a number of frequencies.



Weighting factor: This factor specifies the relative weight of this aim function as compared to additional aim functions.

Polarisation: The values in this box can be used to specify which component of the gain should be optimised.

Treatment for multiple blocks: The options in this box are used to specify the method to combine the local aim functions into global aim functions.

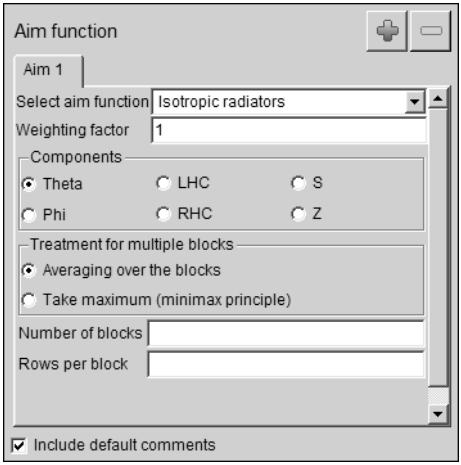
Number of blocks: This parameter indicates the number of blocks in the output file from FEKO, that are to be read. For each block $i = 1 \ldots N_B$ the first row is read and the gain/directivity (g_i) in dB is extracted.

The local aim function is given by $f_i = -g_i$. The minus sign ensures that a minimisation of the aim function maximises the gain.

This aim function is selected by using the keyword `GAIN` or `GEWINN` in the `*.opt` file.

Isotropic radiators

This aim function allows the user to design an antenna with the best possible isotropic gain. Here one can optimise for more than one frequency by using the frequency blocks from the output file from FEKO in the aim function.



Weighting factor: This factor specifies the relative weight of this aim function as compared to additional aim functions.

Components: The components of the far field that should be isotropic.

Treatment for multiple blocks: The options in this box are used to specify the method to combine the local aim functions into global aim functions.

Number of blocks: This parameter indicates the number of blocks in the output file from FEKO, that are to be read.

Rows per block: This must be set to the number of rows that are to be read from each block, i.e. the number of discrete points (ϑ_j, φ_j) when varying the angle.

The keyword `OMNIDIRECTIVITY` or `RUNDSTRAHLUNG` indicates that this aim function should be used.

For each block $i = 1 \ldots N_B$ the roughness in dB is determined according to the relation

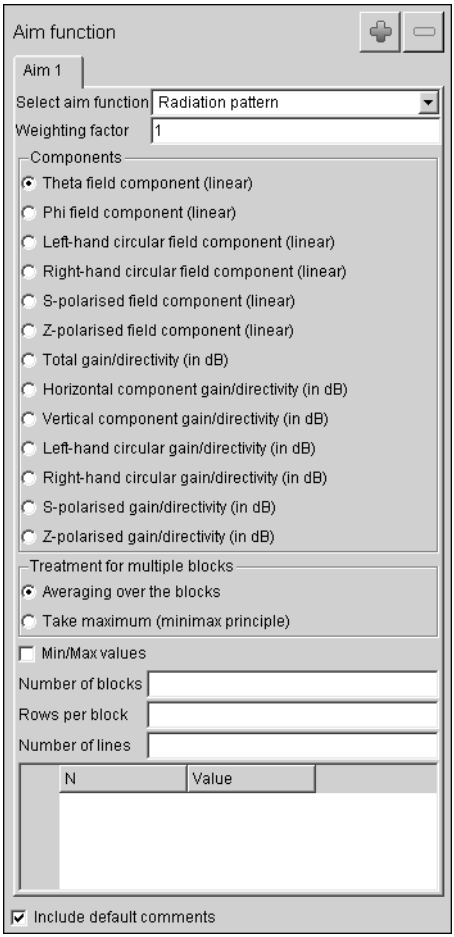
$$f_i = 20 \cdot \log_{10} \left[\frac{\max_j |E_{\vartheta,\varphi}(\vartheta_j, \varphi_j)|}{\min_j |E_{\vartheta,\varphi}(\vartheta_j, \varphi_j)|} \right] \quad .$$

Here j scans through the values $1 \ldots N_S$. Using this error function, the local aim function is related to the maximum roughness.

Radiation pattern

This aim function can be used to optimise the radiation pattern according to an arbitrary shape specified by the user. One can either optimise directly for a specific shape, or one can specify a band and optimise e.g. for a main beam directivity greater than 10 dBi, and a side lobe level of less than -20 dBi.

As for the isotropic radiator one can optimise for more than one frequency by using the frequency blocks from the output file from FEKO in the aim function.



Weighting factor: This factor specifies the relative weight of this aim function as compared to additional aim functions.

Components: The component of the radiation pattern that should be optimised.

Treatment for multiple blocks: The options in this box are used to specify the method to combine the local aim functions into global aim functions.

Min/Max Values: This box should be checked if a range of values should be used as the optimisation target. This is useful for specifying a minimum and maximum acceptable value. Note that this includes also the case where e.g. the gain should just be smaller than a certain value (i.e. maximum specified, but no minimum), just set the minimum gain to e.g. -999.99 dB.

Number of blocks: This parameter indicates the number of blocks in the output file from FEKO, that are to be read. For each block $i = 1 \dots N_B$ the first row is read and the gain/directivity g_i in dB is extracted.

Rows per block: This must be set to the number of rows that are to be read from each block, i.e. the number of discrete points (ϑ_j, φ_j) when varying the angle.

Number of lines: This parameter specifies the number of aim points that will be specified in the table below. The parameter N is the line number within each block. Either the target value or the range of values (see parameter *Min/Max values above* for each line should be entered into the table. These values are either in dB (optimisation for gain/directivity), or represent the normalised radiation pattern in the range $0 \dots 1$.

Regarding the aim function used by FEKO, we have to distinguish several cases. In the following, $P(j)$ is the user specified aim radiation pattern with j in the range $1 \dots N_S$. When the parameter *Min/Max values* is selected these are denoted with $P_{\min}(j)$ and $P_{\max}(j)$ in the following. OPTFEKO automatically uses a linear interpolation for points in between the N_P specified points. Should $P(1)$ or $P_{\min}(1)$, $P_{\max}(1)$ not be defined by the user, OPTFEKO assumes a value of zero (both in the linear case, also for gain/directivity in dB). This might not be correct, thus it is advisable to always specify the desired values for the first angle.

The error function depends on the type of optimisation required.

- Direct optimisation of a field component for the case where *Min/Max values* is unchecked. The error function is given by

$$f_i = \frac{1}{N_S} \sum_{j=1}^{N_S} \left| \frac{|E_{\vartheta,\varphi}(\vartheta_j^i, \varphi_j^i)|}{E_{\max}^i} - P(j) \right|$$

  $E_{\vartheta,\varphi}(\vartheta_j^i, \varphi_j^i)$ is the electric field strength for the j^{th} far field line in the i^{th} block.

  $E_{\max}^i$ represents the maximum field strength amplitude (of the component which is selected) in the i^{th} far field block, i.e. $E_{\max}^i = \max_j |E_{\vartheta,\varphi}(\vartheta_j^i, \varphi_j^i)|$ which is required to normalise the field strength to give the normalised radiation pattern.

- Direct optimisation of a field component for the case where *Min/Max values* is checked. The error function is given by

$$f_i = \frac{1}{N_S}\sum_{j=1}^{N_S}\left[\Gamma\left(\frac{|E_{\vartheta,\varphi}(\vartheta_j^i,\varphi_j^i)|}{E_{\max}^i} - P_{\max}(j)\right) + \Gamma\left(P_{\min}(j) - \frac{|E_{\vartheta,\varphi}(\vartheta_j^i,\varphi_j^i)|}{E_{\max}^i}\right)\right]$$

  with the thresholding function

$$\Gamma(x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases} .$$

- Direct optimisation of a gain/directivity for the case where *Min/Max values* is unchecked. The error function is given by

$$f_i = \frac{1}{N_S}\sum_{j=1}^{N_S} \mid g_{h,v,t}(\vartheta_j^i,\varphi_j^i) - P(j) \mid$$

  $g_{h,v,t}(\vartheta_j^i,\varphi_j^i)$ is the gain or directivity in dB (the component which is selected by M) for the j^{th} far field line in the i^{th} block.

- Direct optimisation of a gain/directivity for the case where *Min/Max values* is checked. The error function is given by

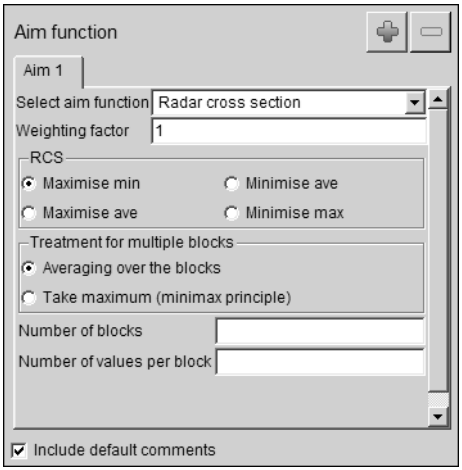
$$f_i = \frac{1}{N_S}\sum_{j=1}^{N_S}\left[\Gamma\left(g_{h,v,t}(\vartheta_j^i,\varphi_j^i) - P_{\max}(j)\right) + \Gamma\left(P_{\min}(j) - g_{h,v,t}(\vartheta_j^i,\varphi_j^i)\right)\right]$$

  with the same thresholding function $\Gamma(x)$ as defined above.

The key word `RADIATION_PATTERN` or `RICHTDIAGRAMM` indicates that this aim function is used.

Radar cross section (RCS)

With this aim function the RCS σ can be minimised or maximised (either the average value or the minimum / maximum value) over a range of angles and over a range of frequencies (blocks).



Weighting factor: This factor specifies the relative weight of this aim function as compared to additional aim functions.

RCS: The type of RCS optimisation that should be performed, must be selected in this box. See the description of the different aim functions below.

Treatment for multiple blocks: The options in this box are used to specify the method to combine the local aim functions into global aim functions.

Number of blocks: This parameter indicates the number of blocks in the output file from FEKO, that are to be read. For each block $i = 1 \ldots N_B$ the first row is read and the gain/directivity g_i in dB is extracted.

Number of values per block: This must be set to the number of rows that are to be read from each block, i.e. the number of discrete points (ϑ_j, φ_j) when varying the angle.

The aim function depends on the choice of the *RCS* box above. For the other options, the 'per block' aim function is given by

- Maximise the minimum RCS

$$f_i = -\max_{j=1\ldots N_S} \sigma(\vartheta_j^i,\, \varphi_j^i)$$

- Maximise the average RCS

$$f_i = \frac{1}{N_S} \cdot \sum_{j=1}^{N_S} \sigma(\vartheta_j^i,\, \varphi_j^i)$$

- Minimise the average RCS

$$f_i = -\frac{1}{N_S} \cdot \sum_{j=1}^{N_S} \sigma(\vartheta_j^i, \varphi_j^i)$$

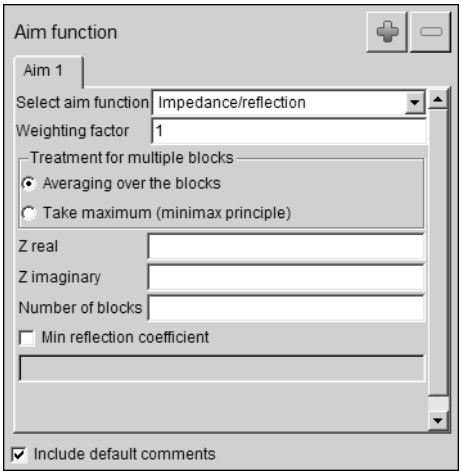
- Minimise the maximum RCS

$$f_i = \max_{j=1\ldots N_S} \sigma(\vartheta_j^i, \varphi_j^i)$$

This aim function is chosen by using the keyword `RCS`.

Impedance / reflection factor

With this aim function, the input reflection factor of an antenna can be minimised. The optimisation can be broad band, by examining a number of frequencies at the same time. One can also optimise the reflection factor of multiple different ports simultaneously.



Weighting factor: This factor specifies the relative weight of this aim function as compared to additional aim functions.

Treatment for multiple blocks: The options in this box are used to specify the method to combine the local aim functions into global aim functions.

Z real: The real part of the target impedance.

Z imaginary: The imaginary part of the target impedance.

Number of blocks: This parameter indicates the number of blocks in the output file from FEKO, that are to be read.

Min reflection coefficient: Select this to use the option to limit values that have a sufficiently good match, to avoid that these are optimised further rather than focusing on the optimisation of other impedance values with a worse match. The minimum value should be entered below this box.

The assignment of the aim function in the `*.opt` file is selected using the keyword `IMPEDANCE` or `IMPEDANZ`. The error function is defined as

$$f_i = \left| \frac{Z_i - Z_{soll}}{Z_i + Z_{soll}} \right|$$

If the *Min reflection coefficient* option is used, an additional value for the minimum reflection coefficient $\Gamma_{\min}^{\mathrm{dB}}$ in dB is specified. The modified algorithm to compute the aim function is as follows.

For all impedance values $i = 1 \ldots N_B$ define the reflection coefficient

$$\Gamma_i = \left| \frac{Z_i - Z_{soll}}{Z_i + Z_{soll}} \right|$$

and then truncate to $\Gamma_{\min} = 10^{\frac{\Gamma_{\min}^{\mathrm{dB}}}{20}}$ (i.e. the corresponding linear desired reflection coefficient) in the aim function:

$$f_i = \max(\Gamma_i, \Gamma_{\min})$$

The reason for doing this is to avoid that one impedance value (such as one port or the impedance at one frequency) which has already an acceptable match is being optimised further and further, instead of focusing on the optimisation of other impedance values where the desired match has not yet been obtained.

For an optimisation where the SWR (or VSWR) should remain smaller than a certain value s, this feature of specifying $\Gamma_{\min}^{\mathrm{dB}}$ can be used. A penalty is added to the aim function if the SWR is larger than s, but nothing is added for values smaller than s. In order to compute the value $\Gamma_{\min}^{\mathrm{dB}}$ that must be specified in the *.opt file from the maximum standing wave ratio s use the relationship

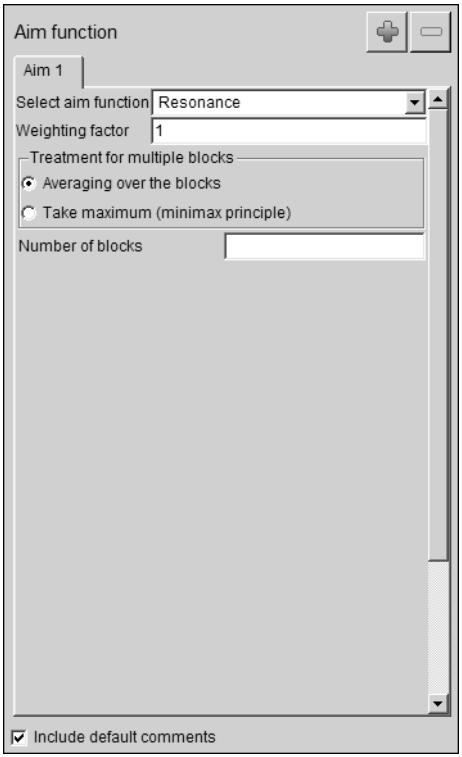
$$\Gamma_{\min}^{\mathrm{dB}} = 20 \log = 20 \log \frac{s-1}{s+1}$$

(e.g. for a SWR limit of $s = 2.5$ we obtain $\Gamma_{\min}^{\mathrm{dB}} = -7.36\,\mathrm{dB}$).

Apart from the value of the aim function Z, the real part and the imaginary part of the input impedance Z_i are written to the log file (the latter only for the first block).

Resonance

This aim function is used to tune an antenna / circuit / resonator to resonance according to the criterion that the imaginary part of the input impedance should be zero.



Weighting factor: This factor specifies the relative weight of this aim function as compared to additional aim functions.

Treatment for multiple blocks: The options in this box are used to specify the method to combine the local aim functions into global aim functions.

Number of blocks: This parameter indicates the number of blocks in the output file from FEKO, that are to be read.

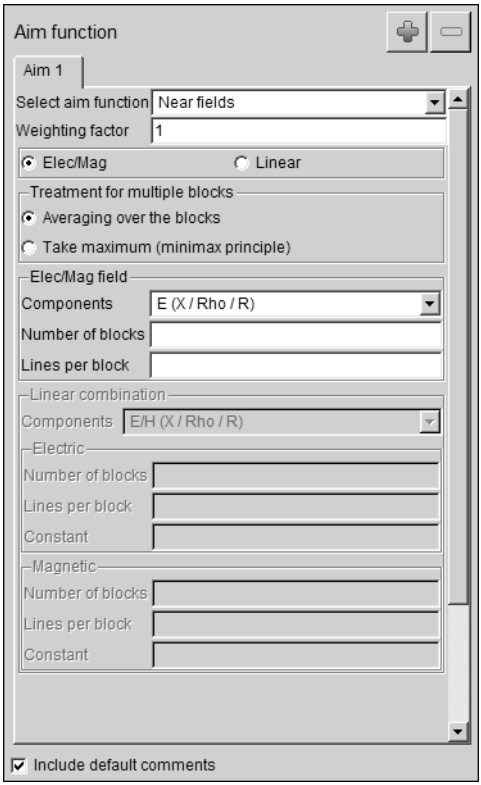
The keyword `RESONANCE` or `RESONANZ` is used in the `*.opt` file to indicate that this keyword should be used. The aim function is specified as

$$f_i = |\operatorname{Im}[Z_i]|$$

with Z_i being the input impedance of the i^{th} block.

Maximisation or minimisation of near fields

This aim function enables one to maximise the electric or magnetic field strength in the near field. This is particularly useful for a transmitting antenna that radiates constant power. In addition any linear combination — with arbitrary proportionality factors — of the electrical and magnetic fields can be maximised (or minimised by using negative values of the proportionality factors).



Weighting factor: This factor specifies the relative weight of this aim function as compared to additional aim functions.

Elec/Mag: Select this option if either the electric OR magnetic field should be optimised.

Linear: Select this option if a linear combination of the electric and magnetic fields should be optimised.

Treatment for multiple blocks: The options in this box are used to specify the method to combine the local aim functions into global aim functions.

Number of blocks: This parameter indicates the number of blocks in the output file from FEKO, that are to be read. Note that different values for this parameter can be set for the electric or magnetic fields.

Lines per block: The number of samples in each block that should be read. Note that different values for this parameter can be set for the electric or magnetic fields.

Components: The field component to be optimised is selected from this list. Note that the options are dependent on the coordinate system that the near fields are calculated in. The options of the field component direction are given inside round brackets on each line. The first option is for Cartesian coordinates, the second for cylindrical and the third for spherical.

The magnitude of the field strength is evaluated in each case. N_B gives the number of blocks, containing electric or magnetic near field strength values, to read and N_S gives the number of lines to read from each of these blocks.

For this case, the aim function f_i is defined as

$$f_i = -\frac{1}{N_S} \cdot \sum_{j=1}^{N_S} |E_{i,j}|$$

(using the magnetic near field $|H_{i,j}|$ when using the magnetic field). The negative sign ensures that a minimisation of the aim function maximises the near field.

As before, only the magnitude of the field strength is evaluated in each case. N_B^E gives the number of electric near field blocks to read and N_S^E the number of lines to read from each of these blocks. Similarly N_B^H gives the number of magnetic near field blocks and N_S^H the number of lines for each block.

The parameters F^E and F^H are arbitrary proportionality constants that may also be negative. The error function is calculated separately for the Electric and Magnetic fields. The aim function is the sum of the aim function for the individual error functions, i.e. $Z = Z^E + Z^H$. Note that the number of blocks of the electric and magnetic fields does not need to be the same. For the electric fields the aim function is given by

$$f_i^E = -\frac{F^E}{N_S^E} \cdot \sum_{j=1}^{N_S^E} |E_{i,j}|$$

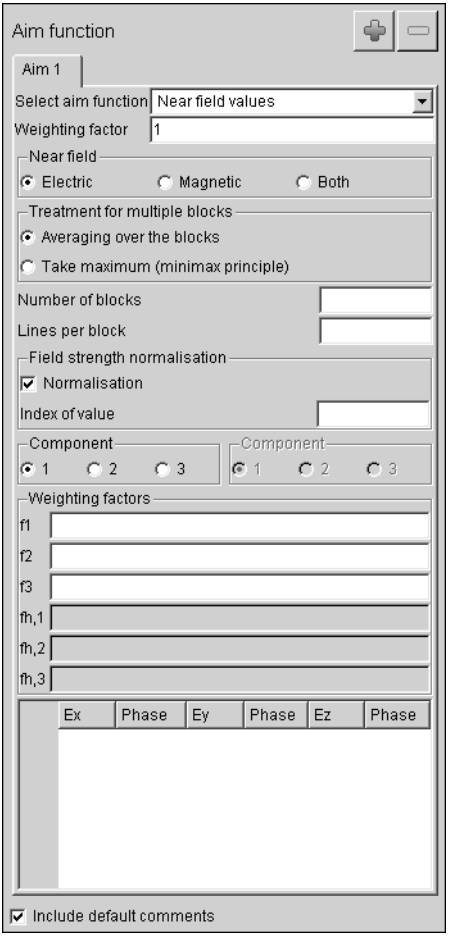
And for the magnetic fields it is given by

$$f_i^H = -\frac{F^H}{N_S^H} \cdot \sum_{j=1}^{N_S^H} |H_{i,j}|$$

The keyword `NEARFIELD` or `NAHFELD` is used to specify this aim function.

Near field values

This aim function enables one to optimise either the electric or the magnetic near field or both with respect to certain defined field strength values. Since the absolute field strength is just scaling linear with the amplitude of the excitation, this extra degree of freedom can be removed by activating an optional normalisation of the near fields with respect to one point (whose amplitude after normalisation will then be one and then phase will be zero). But one can also optimise for absolute field strength values in V/m (electric field) or A/m (magnetic field).



Weighting factor: This factor specifies the relative weight of this aim function as compared to additional aim functions.

Near field: Optimisation of electric fields, magnetic fields or both can be selected here.

Treatment for multiple blocks: The options in this box are used to specify the method to combine the local aim functions into global aim functions.

Number of blocks: This parameter indicates the number of blocks in the output file from FEKO, that are to be read.

Lines per block: The number of samples to read from each block.

Normalisation Field Strength: This number indicates which field strength value to use for the normalisation.

Component: The field component that should be used for normalisation. See the comments below.

Weighting factors: These factors determine the relative weighting of the components of the field. See the discussion below.

Aim value table: The table is used to specify the aim values for the values.

The optimisation parameters follow the keyword `NEARFIELD_VALUES` or `NAHFELD_WERTE`.

N_B gives the number of blocks containing electric or magnetic near field strength values to be read, and N_S gives the number of lines to be read from each of these blocks. The value N_{norm} must be in the range $1 \ldots N_B\, N_S$ and indicates which field strength value shall be used for the normalisation.

Each field strength vector (either electric $\vec{E}$ or magnetic $\vec{H}$) has three components E_1, E_2, E_3 or H_1, H_2, H_3, respectively. For instance, in a Cartesian coordinate system the assignment is $E_1 = E_x$, $E_2 = E_y$ and $E_3 = E_z$, but in a spherical coordinate system one would have $E_1 = E_r$, $E_2 = E_\vartheta$, $E_3 = E_\varphi$. The value of N_{comp}, which must be 1, 2, or 3, determines which component of the N_{norm}th field strength value shall be used for the normalisation. When normalisation has been switched off, the component no longer needs to be specified.

The weighting parameters f_1, f_2, and f_3 are discussed below when describing the aim function.

Let $E_{k,i,j}$ be the kth (with $k = 1, 2, 3$) component of the electric field strength in the ith block (with $i = 1 \ldots N_B$) and the jth line in this block (with $j = 1 \ldots N_S$), then we define normalised field strength values as

$$e_{k,i,j} = \frac{E_{k,i,j}}{E_{norm}}$$

where E_{norm} is the N_{comp}th component of the N_{norm}th field strength value that has been selected for the normalisation (this is a complex quantity with amplitude and phase). When normalisation has been switched off, then in the above equation we normalise with $E_{norm} = 1\frac{\mathrm{V}}{\mathrm{m}}$ (i.e. basically $e_{k,i,j}$ is then equal to $E_{k,i,j}$ but without dimension). For the magnetic near field with normalisation switched off we use $H_{norm} = 1\frac{\mathrm{A}}{\mathrm{m}}$.

The aim function f_i for the optimisation is then defined as

$$f_i = \frac{1}{N_S} \cdot \sum_{j=1}^{N_S} \left[f_1 \left| e_{1,i,j} - e^{aim}_{1,i,j} \right|^2 + f_2 \left| e_{2,i,j} - e^{aim}_{2,i,j} \right|^2 + f_3 \left| e_{3,i,j} - e^{aim}_{3,i,j} \right|^2 \right] \quad .$$

The values $e^{aim}_{k,i,j}$ are the desired field values, and they must be specified in the table on the panel. There are in total $N_B\, N_S$ lines with all the field values, and each line has 6 columns with magnitude and phase (in degree) of the 3 normalised field strength components $e_{1,j,k}$, $e_{2,j,k}$ and $e_{3,j,k}$ (the inner loop is the one over j, i.e. the values appear in the same order as they would be in the output file):

$$|e_{1,i,j}| \quad \text{phase}(e_{1,i,j}) \quad |e_{2,i,j}| \quad \text{phase}(e_{2,i,j}) \quad |e_{3,i,j}| \quad \text{phase}(e_{3,i,j})$$

In the line N_{norm} of this table, the component N_{comp} (which we are normalising to) must be 1 in amplitude and 0 in phase.

So far only the case for the electric field has been discussed (normalisation, aim function, specification of values in the `*.opt` file), but the case of the magnetic field is analogous, one must just replace E by H and e by h in the formulas above.

If one wants to enforce to have only E_z, but no E_x and E_y components, then all one has to do is to change the two factors f_1 and f_2 (which in a Cartesian coordinate system refer to the x and y components) both to 1.0, so that these components are included in the aim function.

Both electric and magnetic fields can be optimised for by selecting the appropriate button. When optimising both field simultaneously, the meaning of N_B, N_S and N_{norm} is still the same as when optimising only E or H, but for the component N_{comp} one can now specify this separately for the electric and magnetic field using $N_{e,comp}$ or $N_{h,comp}$, respectively.

There are now factors f, which are used in the error function

$$\begin{aligned} f_i = \frac{1}{N_S} \cdot \sum_{j=1}^{N_S} \Big[& f_{e,1} \left| e_{1,i,j} - e^{aim}_{1,i,j} \right|^2 + f_{e,2} \left| e_{2,i,j} - e^{aim}_{2,i,j} \right|^2 + f_{e,3} \left| e_{3,i,j} - e^{aim}_{3,i,j} \right|^2 \\ & + f_{h,1} \left| h_{1,i,j} - h^{aim}_{1,i,j} \right|^2 + f_{h,2} \left| h_{2,i,j} - h^{aim}_{2,i,j} \right|^2 + f_{h,3} \left| h_{3,i,j} - h^{aim}_{3,i,j} \right|^2 \Big] \quad . \end{aligned}$$

The table with the required field strength values has now also both E and H field values in one line.

11.3 Running OPTFEKO

Firstly the `*.pre` and `*.opt` files must be created as discussed above. During optimisation new `*.pre` input files are continuously created by adding the string `_opt_` and a sequentially incremented number to the file name. When, for example, the files `dipole.pre` and `dipole.opt` have been created, OPTFEKO is run with the command:

```
optfeko dipole
```

On the command line the following parameters can be added:

`-r` The `*.pre` and `*.out` are deleted after each analysis. This saves disk space.

`-R` Same as "-r", but the files of the optimum solution are kept.

`-z` The value of the aim function is calculated for one existing file — no optimisation is done. This is mostly used for debugging.

`--resume` Resumes an optimisation process that has been stopped, provided that the files `*.sta` and `*.bat` are still available (for example, the previous optimisation has been stopped by pressing <Ctrl><C> or due to a power failure or a FEKO error etc.). For Windows this option must follow right after the filename, i.e. one must call

```
optfeko file --resume -r
```

and not

```
optfeko file -r --resume
```

`--runfeko-options` After this option one can specify additional options which will be used in the launcher RUNFEKO for the FEKO kernel. For instance in order to use the parallel FEKO solver during the optimisation one can use the command

```
optfeko file --runfeko-options -np 2
```

or also

```
optfeko file --runfeko-options -np 2 --machines-file m
```

where `m` indicates the machines file. For a remote execution of the FEKO runs during the optimisation on another host, the suitable command would be

```
optfeko file --remote-host hostname
```

See section 8.2 for a list of all such RUNFEKO options.

Information on the optimisation process is stored in a log file with the extension `.log` — in the example above the filename will be `dipole.log`.

It should be noted that when using the remote launching facility (see section 8.2.3), the actual optimisation is done on the local machine, only the FEKO kernel runs (which are the time and memory consuming part) are done on the remote machine.

11.4 An example using OPTFEKO

In figure 11-1 a dipole antenna in front of a reflector is shown. The gain is to be optimised. The bent angle α of the dipole and the distance a between the dipole and the reflector are given as optimisation parameters. The input files for this optimisation example can be found in your FEKO examples directory under `...\utils\optfeko\dipole.*`.

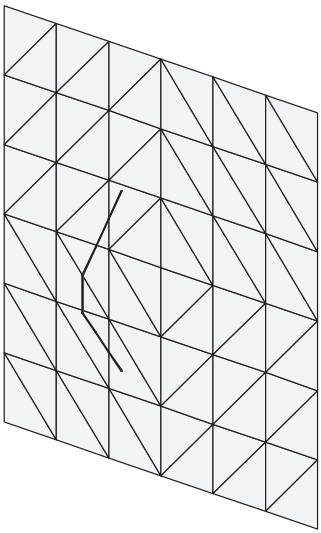


Figure 11-1: Bent dipole in front of a reflector

As a first step, a discrete point search is done by systematically varying the two optimisation parameters to find the region of the minimum.

The gain (the negative aim function) can be displayed graphically. In figure 11-2 one can see a clear maximum in the area $\alpha \approx 10°$ and $\frac{a}{\lambda} \approx 0.8$.

The general "position" of the optimum is now known. A new search can now begin with e.g. the simplex method, by using the input found in `dipole.opt`.

The extract from the log file reproduced below shows that the procedure converges for $\alpha \approx 7°$ and $\frac{a}{\lambda} \approx 0,78$

```
========================== Optimisation and Analysis ==========================

No.  #alpha       #a            Aim fct.     Penaltyfct.  other spec.
  1  2.0000e+01  8.0000e-01  -7.1538e+00  0.0000e+00   1 -7.1538e+00
  2  2.8693e+01  8.0776e-01  -7.0386e+00  0.0000e+00   1 -7.0386e+00
  3  2.2329e+01  8.2898e-01  -7.0007e+00  0.0000e+00   1 -7.0007e+00
  4  2.6364e+01  7.7879e-01  -7.0176e+00  0.0000e+00   1 -7.0176e+00
```

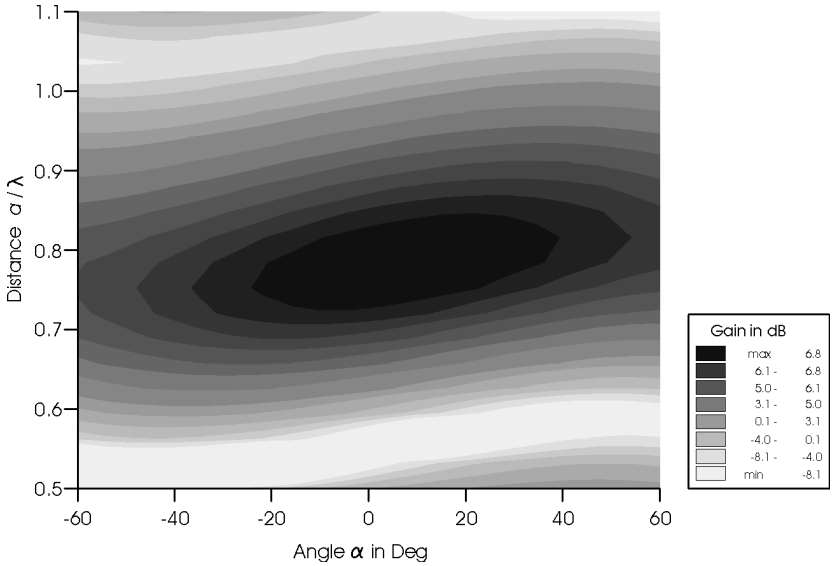


Figure 11-2: Gain as a function of the optimisation parameters

```
 5  1.7671e+01  7.7102e-01  -7.1202e+00  0.0000e+00   1 -7.1202e+00
 6  1.1307e+01  7.9224e-01  -7.2145e+00  0.0000e+00   1 -7.2145e+00
 7  1.3636e+01  8.2121e-01  -7.0740e+00  0.0000e+00   1 -7.0740e+00
 8  1.5653e+01  7.9612e-01  -7.1914e+00  0.0000e+00   1 -7.1914e+00
 9  1.2471e+01  8.0672e-01  -7.1658e+00  0.0000e+00   1 -7.1658e+00
10  1.4489e+01  7.8163e-01  -7.1970e+00  0.0000e+00   1 -7.1970e+00
11  1.0142e+01  7.7775e-01  -7.2171e+00  0.0000e+00   1 -7.2171e+00
12  6.9600e+00  7.8835e-01  -7.2225e+00  0.0000e+00   1 -7.2225e+00
13  5.7953e+00  7.7386e-01  -7.2218e+00  0.0000e+00   1 -7.2218e+00
14  2.6133e+00  7.8447e-01  -7.2150e+00  0.0000e+00   1 -7.2150e+00
15  3.7780e+00  7.9896e-01  -7.1728e+00  0.0000e+00   1 -7.1728e+00
16  8.1247e+00  8.0284e-01  -7.1769e+00  0.0000e+00   1 -7.1769e+00
17  9.1333e+00  7.9029e-01  -7.2205e+00  0.0000e+00   1 -7.2205e+00
18  7.5423e+00  7.9560e-01  -7.2053e+00  0.0000e+00   1 -7.2053e+00
19  8.5510e+00  7.8305e-01  -7.2270e+00  0.0000e+00   1 -7.2270e+00
20  6.3777e+00  7.8111e-01  -7.2282e+00  0.0000e+00   1 -7.2282e+00
21  7.9686e+00  7.7581e-01  -7.2214e+00  0.0000e+00   1 -7.2214e+00
22  5.7953e+00  7.7386e-01  -7.2218e+00  0.0000e+00   1 -7.2218e+00
23  4.2043e+00  7.7917e-01  -7.2255e+00  0.0000e+00   1 -7.2255e+00
24  4.7867e+00  7.8641e-01  -7.2207e+00  0.0000e+00   1 -7.2207e+00
25  7.4643e+00  7.8208e-01  -7.2281e+00  0.0000e+00   1 -7.2281e+00
26  6.6688e+00  7.8473e-01  -7.2268e+00  0.0000e+00   1 -7.2268e+00
27  7.1732e+00  7.7846e-01  -7.2266e+00  0.0000e+00   1 -7.2266e+00
28  6.0865e+00  7.7749e-01  -7.2266e+00  0.0000e+00   1 -7.2266e+00
```

```
 29  5.2910e+00  7.8014e-01  -7.2274e+00  0.0000e+00   1 -7.2274e+00
 30  5.5822e+00  7.8376e-01  -7.2262e+00  0.0000e+00   1 -7.2262e+00
 31  6.9210e+00  7.8159e-01  -7.2283e+00  0.0000e+00   1 -7.2283e+00
 32  6.5232e+00  7.8292e-01  -7.2279e+00  0.0000e+00   1 -7.2279e+00
 33  6.7754e+00  7.7978e-01  -7.2279e+00  0.0000e+00   1 -7.2279e+00
 34  7.3187e+00  7.8027e-01  -7.2277e+00  0.0000e+00   1 -7.2277e+00
 35  7.4643e+00  7.8208e-01  -7.2281e+00  0.0000e+00   1 -7.2281e+00
 36  7.0666e+00  7.8341e-01  -7.2279e+00  0.0000e+00   1 -7.2279e+00
 37  7.1927e+00  7.8184e-01  -7.2282e+00  0.0000e+00   1 -7.2282e+00
 38  6.9938e+00  7.8250e-01  -7.2282e+00  0.0000e+00   1 -7.2282e+00

Optimisation finished (standard dev. small enough: 5.7735e-05)

Result destination vector: #alpha                          = 6.9209918e+00
                           #a                              = 7.8159398e-01

Minimal value of the aim function (at no. 31): -7.2283000e+00
no. of the last analysis: 38
```

12 The program TIMEFEKO

12.1 Description

With the program TIMEFEKO electromagnetic scattering problems can be solved in the time domain. It is based on the program FEKO, that does the calculation in the frequency domain, and an FFT algorithm, that transforms the data to the time domain.

For the excitation a number of different pulses have been defined and stored in the function library. The functions available at present are described in section 12.3.1.

The program TIMEFEKO is constructed in such a way that all the data in the output file `*.out` from FEKO are transformed, i.e. in the input file the cards have to be specified in the correct way to ensure that the correct data is transformed. Information on the correct card definitions is given in the following section.

The program TIMEFEKO uses two input files,[11] that have to be created by the user.

12.2 The *.pre input file

The input file with extension `*.pre` is the normal input file for PREFEKO, in which the frequency *has* to be expressed symbolically. While the program TIMEFEKO is running, the frequency is constantly changed. TIMEFEKO generates new `*.pre` files in which the actual numerical value of the frequency is assigned. The user may use the DEFINED function to assign a numerical value to the frequency variable `#freq` in the `*.pre` — this is useful for running PREFEKO and viewing the geometry in POSTFEKO (see section 12.6).

The FR card *must* use the frequency variable `#freq`. The number of frequencies to be examined *must* be one (NFREQ=1).

Example:

```
** Set the frequency
FR       1                              #freq
```

In the input file `*.pre` all the desired output parameters in their respective cards must be set (for example, FF card, FE card, OS card, ...). Only the data in the output file `*.out` can be transformed.

Note:

The program TIMEFEKO does not check the `*.pre` file. Special care should be taken to ensure that the segmentation parameters for the required frequency interval are fine enough for the program FEKO (see IP card, section 9.2.19). If they are not strictly adhered to, the program can terminate with an error message.

[11] The files have the same filename, but with `*.pre` and `*.tim` extensions.

It is also possible to vary the segmentation for certain applications (far field, near field) to save memory and computation time. To do this one may use the variable `#freq` (which is constantly changed by TIMEFEKO) in the definition of the segmentation parameters.

Example:

```
** Define some constants
#maxfreq = 250.0e+08
#minlambda = #c0/#maxfreq

** Define the edge length  -  note the use of #freq
#edgelen = (2.0-#freq/#maxfreq)*#minlamdba/4.0

** Set the segmentation parameters
IP                                        #edgelen
```

12.3 The *.tim input file

In the file `*.tim` the pulse form, position and the point in time are assigned. This file consists of a number of sections which are optional at present.

- Set the pulse form with characteristic value and the time shift
- Set the size of the highest frequency and the sampling points
- Set the normalisation (normalise time to the speed of light)
- Set whether the output is written to the output file
- Set the time points

In the `*.tim` file, empty or comment lines starting with `**` are allowed. The parameters need not be entered in any particular column, but they have to be in the correct order.

The keywords used in the `*.tim` file exist in German and English (for example `ANREGUNG` and `EXCITATION`). Both versions of each keyword will be given in the discussion below. TIMEFEKO will recognise keywords in either language, independent of the language selected by the environment variable `FEKOLANG`.

12.3.1 Defining the pulse form

The assignment of the pulse form is necessary. Each pulse has a predefined name and has particular characteristic parameters. The parameters must be assigned absolute values and not normalised values. The amplitude factor u_0 is 1. It can be changed by using the Ax card (in the file `*.pre`) with the appropriate amplitude value.

Time shifting t_0: indicates the time (in seconds) that the pulses are delayed (see the shift in figure 12-1). The time shift should be such that the excitation of the structure 0 at $t = 0$ is approximately 0. Since both the time and frequency domain data are continuous this is not strictly required, but it simplifies working with the results.

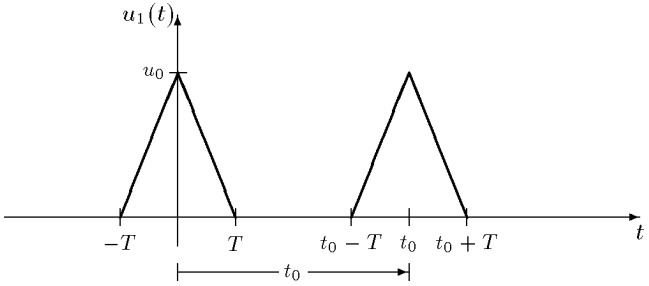


Figure 12-1: Time function $u_1(t)$ shifted by t_0

The following pulses are available:

1. Gaussian pulse `GAUSS / GAUSS` (see figure 12-2)

$$u_1(t) = u_0 \, \mathrm{e}^{-a^2(t-t_0)^2} \tag{12-1}$$

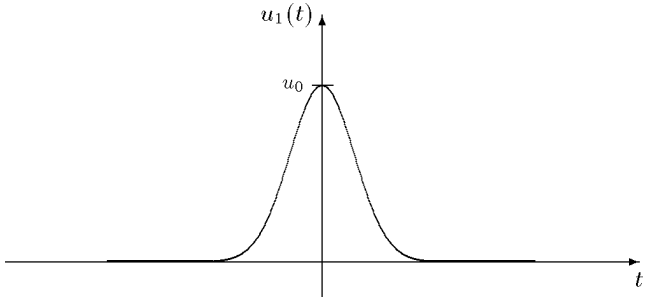


Figure 12-2: Gaussian pulse

Example:

```
** Pulse form
GAUSS
**      t0          Exponent a
    2.0E-08          3.0E+08
```

2. Triangular pulse `DREIECK / TRIANGLE` (see figure 12-3)

$$u_1(t) = \begin{cases} u_0 \left(1 - \dfrac{|t-t_0|}{T}\right) & \text{for} \quad |t-t_0| \leq T \\ 0 & \text{for} \quad \text{otherwise} \end{cases} \tag{12-2}$$

Example:

```
** Pulse form
TRIANGLE
**      t0          Impuls Duration T
    2.0E-08             1.0E-08
```

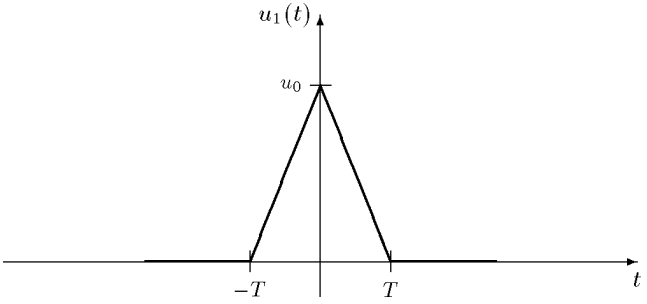


Figure 12-3: Triangular pulse

3. Double exponential pulse `DEXP / DEXP` (see figure 12-4)

$$u_1(t) = \begin{cases} 0 & \text{for} \quad t \le t_0 \\ u_1 \left(1 - \mathrm{e}^{-\frac{t-t_0}{\tau_1}}\right) & \text{for} \quad t_0 \le t \le T + t_0 \\ u_2 \, \mathrm{e}^{-\frac{t-t_0}{\tau_2}} & \text{for} \quad t \ge T + t_0 \end{cases} \tag{12-3}$$

$$u_1 = \frac{u_0}{1 - \mathrm{e}^{-\frac{T-t_0}{\tau_1}}} \qquad u_2 = \frac{u_0}{\mathrm{e}^{-\frac{T-t_0}{\tau_2}}} \tag{12-4}$$

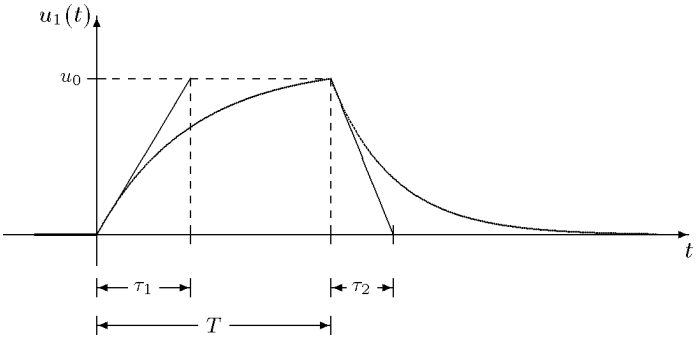


Figure 12-4: Double exponential pulse

Example:

```
** Pulse form
DEXP
**    t0          T                 tau1              tau2
      0.0         10.0E-09          5.0E-9            10.0E-9
```

4. Ramp pulse `RAMPE / RAMP` (see figure 12-5)

$$u_1(t) = \begin{cases} 0 & \text{for} \quad t \le t_1 + t_0 \\ u_0 \left(1 - \dfrac{|t - t_2 - t_0|}{\tau_1}\right) & \text{for} \quad t_1 + t_0 \le t \le t_2 + t_0 \\ u_0 & \text{for} \quad t_2 + t_0 \le t \le t_3 + t_0 \\ u_0 \left(1 - \dfrac{|t - t_3 - t_0|}{\tau_2}\right) & \text{for} \quad t_3 + t_0 \le t \le t_4 + t_0 \\ 0 & \text{for} \quad t \ge t_4 \end{cases} \tag{12-5}$$

$$t_1 = -\frac{T + \tau_1}{2} \qquad t_2 = -\frac{T - \tau_1}{2} \qquad t_3 = \frac{T - \tau_2}{2} \qquad t_4 = \frac{T + \tau_2}{2} \tag{12-6}$$

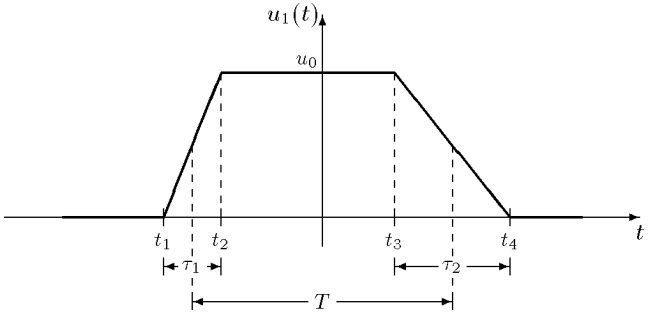


Figure 12-5: Ramp pulse

Example:

```
** Pulse form
RAMP
**   t0          Impuls Duration T     tau1              tau2
   20.0E-09         15.0E-09          5.0E-09          10.0E-09
```

5. Double exponential impulse (second type) `DBLEXP / DBLEXP` (see figure 12-6)

$$u_1(t) = \begin{cases} 0 & \text{for} \quad t \le t_0 \\ u_0 \left(\mathrm{e}^{-\frac{t-t_0}{\tau_1}} - \mathrm{e}^{-\frac{t-t_0}{\tau_2}}\right) & \text{for} \quad t > t_0 \end{cases} \tag{12-7}$$

Example:

```
** Impulse form
DBLEXP
** Time. t0      Parameter tau1     Parameter tau2
     20.0E-9         70.0E-9            5.0E-9
```

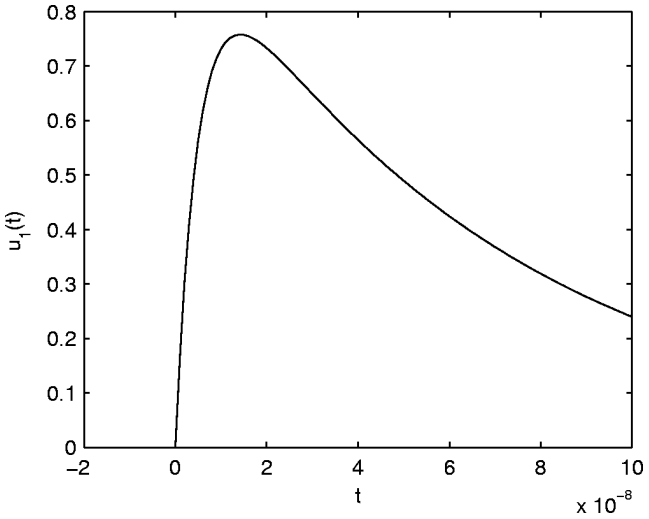


Figure 12-6: Double exponential impulse (second type)

12.3.2 Definition of the frequency block

The upper frequency limit f_{max} and the number of frequency points N are defined in the frequency block `FREQUENZ / FREQUENCY`.

Example:

```
FREQUENCY
** Upper frequency limit          Number of frequency points
          250.0E+06                          34
```

The maximum frequency f_{max} should be large enough such that the whole spectrum of the exciting pulse is covered. For example, for the Gaussian pulse `GAUSS` of section 12.3.1 we find

$$f_{3dB} = \frac{a}{\pi}\sqrt{\ln\sqrt{2}} \approx 0.187\, a$$

with a as defined in equation 12-1. One should select approximately $f_{max} = 4\, f_{3dB}$.

The number of frequency points N is then selected such that the total time T is long enough for the exciting pulse and all included currents, radiated fields, etc. to have decayed. Once the total time has been determined, the number of samples may be determined from the relation

$$\Delta f = \frac{1}{T} = \frac{f_{max}}{(N-1)}$$

or

$$N = 1 + T\, f_{max} \tag{12-8}$$

Finally, let P be the smallest power of 2 which is larger than or equal to $N-1$. (For example: for N=10, P=16; for N=33, P=32; and for N=50, P=64.) Then the time stepping will be $\Delta t = \frac{T}{2P} = \frac{N-1}{2\,f_{max}\,P}$. With the relation $\frac{P}{2} < N-1 \leq P$ we get the bounds

$$\frac{1}{4\,f_{max}} < \Delta t \leq \frac{1}{2\,f_{max}}$$

12.3.3 Definitions of the normalisation

Using the keyword `NORM / NORM`, the time can be normalised to the speed of light in a vacuum c_0. The normalised time then has a unit of lm *(light-metre)*.

$$t_{norm} = t \cdot c_0$$

Example:

```
** Normalising time with respect to the speed of light
NORM
```

12.3.4 Definition of the excitation output

The keyword `ANREGUNG / EXCITATION` indicates whether the time variation of the exciting pulse is to appear in the output file.

Example:

```
** Output the excitation pulse
EXCITATION
```

12.3.5 Definition of a time point

With the keyword `ZEITPUNKTE / POINTS_IN_TIME`, the near fields, surface currents and line currents can be calculated at certain points in time. Using time points has no effect on the calculation of source currents or far fields. If normalisation has been set (`NORM`), then the time points must also be given in normalised form.

Example:

```
** Specifying the time points -- time has been normalised
POINTS_IN_TIME
  5.0   10.0  15.0  20.0
```

12.4 Running TIMEFEKO

Firstly, the input files `*.pre` and `*.tim` have to be created. During execution new input files `*.pre` are continuously generated by adding the string `_tim_` and a sequentially incremented number to the file name. If, for example, the input files `cube.pre` and `cube.tim` have been created, TIMEFEKO is executed with the command

```
timefeko cube
```

On the command line the following parameters can be added:

`-a` Here an FFT is only performed on the already available data

`-r` The `*.pre` and `*.out` files are deleted after each iteration. This saves disk space.

`--runfeko-options`
After this option one can specify additional options which will be used in the launcher RUNFEKO for the FEKO kernel. For instance in order to use the parallel FEKO solver during the time domain analysis one can use the command

```
timefeko file --runfeko-options -np 2
```

or also

```
timefeko file --runfeko-options -np 2 --machines-file m
```

For a remote execution of the FEKO runs during the analysis on another host, the suitable command would be

```
timefeko file --remote-host hostname
```

See section 8.2 for a list of all such RUNFEKO options.

It should be noted that when using the remote launching facility (see section 8.2.3), the actual TIMEFEKO process is executed on the local machine, only the FEKO kernel runs (which are the time and memory consuming part) are done on the remote machine.

12.5 TIMEFEKO output

TIMEFEKO generates different `*.pre` files at the different frequencies. The `*.out` files for all these runs are available if TIMEFEKO was called without the `-r` option. These results are summarised — if they are requested in the `*.pre` file — in `*.ost` (triangle currents), `*.oss` (segment currents), `*.cur` (the currents on voltage sources), `*.far` (far fields), `*.nfe` (near electric fields) and `*.nfh` (near magnetic fields). In each case the results for each frequency is listed one after the other with frequencies separated by a line containing a single #-character.

The time domain results are given in the `*.aus` file for all outputs requested in the `*.pre` file. Currents are requested by the OS card. Note that requesting the current on a large number of structures will result in very large `*.aus` files.

12.6 A TIMEFEKO example

In figure 12-7 an ideal conducting metallic cube with side lengths of 1m is shown. The current in the middle of the front side, the scattered field from the direction of incidence (the incident wave travels in the negative x direction), as well as the excitation pulse are to be calculated.

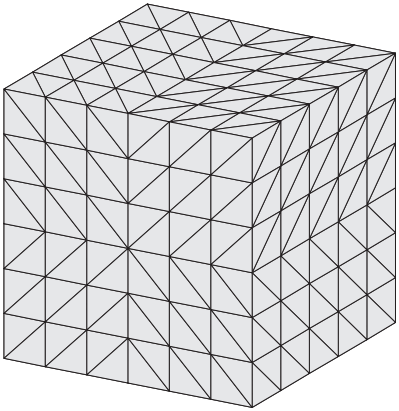


Figure 12-7: Cube with side lengths of 1m

The input file `cube.pre` is reproduced below. (The files are located in the `examples\utils\timefeko` subdirectory of the FEKO installation.)

```
** TIMEFEKO example (*.pre file)
** A metallic cube with side lengths 1m. Only 1/8 of the cube is generated
** explicitly, the rest of the cube is generated by means of symmetry.

** Normally TIMEFEKO will automatically insert the correct required
** frequency value. Use the following construct so that this value
** used by TIMEFEKO will not be overwritten, but we can still display
** the geometry in WinFEKO.
!!if not defined(#freq) then
#freq = 100.0e6
!!endif

** Define some constants
#a = 1           ** side length of the cube
#edgelen = #a/5  ** max. edge length for the triangular patches

** Set the segmentation parameters
IP                                      #edgelen
```

```
** Define the points
DP   P1                          #a/2      0         0
DP   P2                          #a/2      #a/2      0
DP   P3                          #a/2      #a/2      #a/2
DP   P4                          #a/2      0         #a/2
DP   P5                          0         0         #a/2
DP   P6                          0         #a/2      #a/2
DP   P7                          0         #a/2      0

** Create one eigth of the cube (use label 1 for the front plate and
** label 0 for the rest)
LA   1
BP   P1   P2   P3   P4
LA   0
BP   P3   P4   P5   P6
BP   P2   P3   P6   P7

** Mirror around to coordinate planes so that label for front plate
** remains 1, all other surfaces will have label 0)
**   x=0 (yz-plane): only geometric symmetry
**   y=0 (xz-plane): ideal magnetic conducting plane
**   z=0 (xy-plane): ideal electric conducting plane
SY   1    1    0    0    1
CB   2    0
SY   1    0    3    2

** End of the geometry
EG   1    0    1    0    0

** Set the frequency
FR   1    0                   #freq

** Excitation by means of an incident plane wave
A0   0         1    1          1         0         90        0         0

** Surface current density output for surface with label 1
OS   4    1    1

** Calculate the far field only in the direction of incidence
FF   2

** End
EN
```

For this example we have chosen a Gaussian pulse excitation with $a = 3 \times 10^8$. As discussed in section 12.3.2, f_{3dB} is approximately 56 MHz such that we require a maximum frequency of at least 224 MHz. We select a maximum frequency of 250 MHz. The time shift selected for this example is 6 light-metre and the structure dimensions is of the order of 1 metre. Thus we believe that the time response should die out within 40 light-metre or 133 ns. Then equation 12-8 then yields $N = 34$.

The `*.tim` input file `cube.tim` then contains

```
** Timefeko Example (*.tim file)

** Define the Pulse form
GAUSS
** Parameters of the Gaussian pulse
**   Time shift      Exponent
       2.0e-8          3.0e+8

** Define the frequency block
**   Gaussian pulse with a=3.0e+8 1/s, i.e. f_3dB = 0.187*a = 56.2 MHz
**   Choose f_max > 4*f_3dB = 224.9 MHz, use f_max = 250 MHz
**   Total time we want to analyse T = 40 lightmetres = 133.4 ns,
**   i.e. N=1+T*f_max = 34
FREQUENCY
** Upper frequency     Number of Samples
       250.0e+06                34

** Normalise the time to that of the speed of light
NORM

** Output the excitation
EXCITATION
```

The following is an extract from the output file, `cube.aus`

```
          TEMPORAL VARIATION OF EXCITATION NORMALISED TO U_0

    x       y       z
   0.0     0.0     0.0
Time in lm       Value
0.0000000e+000  2.31952283024e-016
3.0916097e-001  8.63275340216e-015
6.1832194e-001  2.65316865993e-013
9.2748292e-001  6.73356755347e-012
```

```
1.2366439e+000  1.41120603102e-010
1.5458049e+000  2.44230818918e-009
1.8549658e+000  3.49040118194e-008
2.1641268e+000  4.11922076322e-007
2.4732878e+000  4.01439061167e-006
2.7824487e+000  3.23064343845e-005

....

      VALUES OF THE SCATTERED ELECTRIC FIELD STRENGTH IN THE FAR FIELD in V
                   Factor e^(-j*BETA*R)/R not considered

   THETA     PHI
  90.00     0.00
  Time in lm          ETHETA              EPHI
0.0000000e+000 -2.98948137392e-005  0.00000000000e+000
3.0916097e-001 -6.77873335003e-005  0.00000000000e+000
6.1832194e-001  4.51882462221e-005  0.00000000000e+000
9.2748292e-001  6.42849147499e-005  0.00000000000e+000
1.2366439e+000 -6.07906709346e-005  0.00000000000e+000
1.5458049e+000 -4.10609464655e-005  0.00000000000e+000
1.8549658e+000  1.46284430530e-004  0.00000000000e+000
2.1641268e+000  3.56160952549e-004  0.00000000000e+000
2.4732878e+000  1.37768474311e-003  0.00000000000e+000
2.7824487e+000  5.67653919004e-003  0.00000000000e+000
3.0916097e+000  1.78441744890e-002  0.00000000000e+000
3.4007707e+000  4.45400653850e-002  0.00000000000e+000
3.7099317e+000  8.92614320231e-002  0.00000000000e+000
4.0190926e+000  1.40179442097e-001  0.00000000000e+000

...


           VALUES OF THE CURRENT DENSITY VECTOR ON TRIANGLES in A/m (no averaging)
number            x/m           y/m           z/m
       1   5.00000E-01  5.55556E-02  5.55556E-02
  Time in lm           JX                  JY                  JZ
0.0000000e+000  0.00000000000e+000  3.80396548298e-009 -2.65565071677e-007
3.0916097e-001  0.00000000000e+000 -3.29242237538e-009  1.93726675808e-006
6.1832194e-001  0.00000000000e+000 -2.60940425868e-009  1.37480675066e-006
9.2748292e-001  0.00000000000e+000  4.56971413492e-009 -9.30586425956e-007
1.2366439e+000  0.00000000000e+000  2.25486652232e-009 -7.87315697551e-007
1.5458049e+000  0.00000000000e+000 -4.83790802156e-009  1.35885783156e-006
1.8549658e+000  0.00000000000e+000 -2.80525051677e-010  1.63413481451e-006

...
```

```
number            x/m           y/m           z/m
       2   5.00000E-01   1.11111E-01   1.11111E-01
  Time in lm           JX                JY                JZ
0.0000000e+000  0.00000000000e+000 -1.18216923283e-008 -1.93459474753e-007
3.0916097e-001  0.00000000000e+000 -9.02561817670e-010  1.95358888266e-006
6.1832194e-001  0.00000000000e+000  1.28360949904e-008  1.34406393148e-006
9.2748292e-001  0.00000000000e+000  4.46200189557e-009 -9.71500429087e-007
1.2366439e+000  0.00000000000e+000 -4.91955007365e-009 -8.71581870796e-007
1.5458049e+000  0.00000000000e+000  9.27286771090e-010  1.21262288818e-006
1.8549658e+000  0.00000000000e+000  3.59541937732e-009  1.49801053069e-006

...

number            x/m           y/m           z/m
       3   5.00000E-01   5.55556E-02   2.22222E-01
  Time in lm           JX                JY                JZ
0.0000000e+000  0.00000000000e+000  9.87039561714e-009 -1.55875015522e-007
3.0916097e-001  0.00000000000e+000 -1.10165720719e-008  1.96750870513e-006
6.1832194e-001  0.00000000000e+000 -3.92248537619e-009  1.37781030380e-006
9.2748292e-001  0.00000000000e+000  1.95306492829e-008 -8.84072619127e-007
1.2366439e+000  0.00000000000e+000  6.20307013697e-009 -7.89929184543e-007
1.5458049e+000  0.00000000000e+000 -1.97321161834e-008  1.24837047986e-006
1.8549658e+000  0.00000000000e+000 -1.39083930925e-009  1.53916294269e-006

...
```

Figure 12-8 shows the response of the excitation $\vec{E}_i(t)$ in the time domain and figure 12-9 the back scattered electric far field E_z.

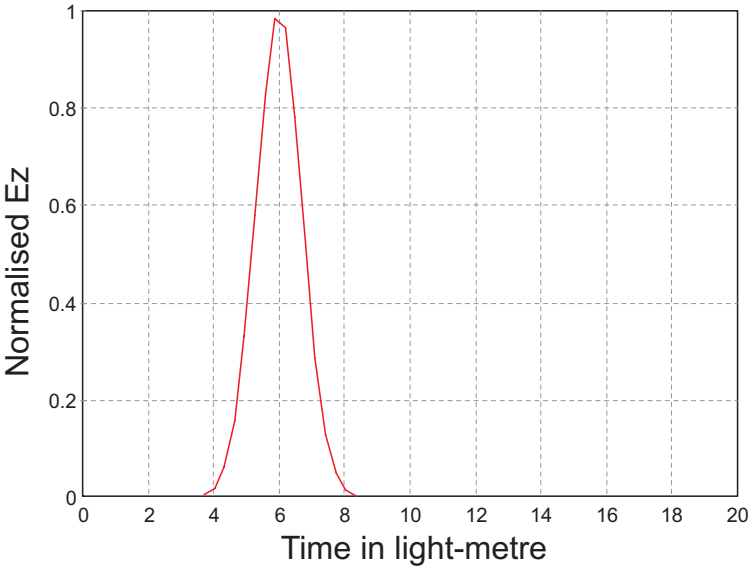


Figure 12-8: Time response of the excitation E_{iz}. The parameters of the Gaussian pulse are: $a = 3.0 \cdot 10^8\ \mathrm{s}^{-1}$, $t_0 = 20\,\mathrm{ns}$.

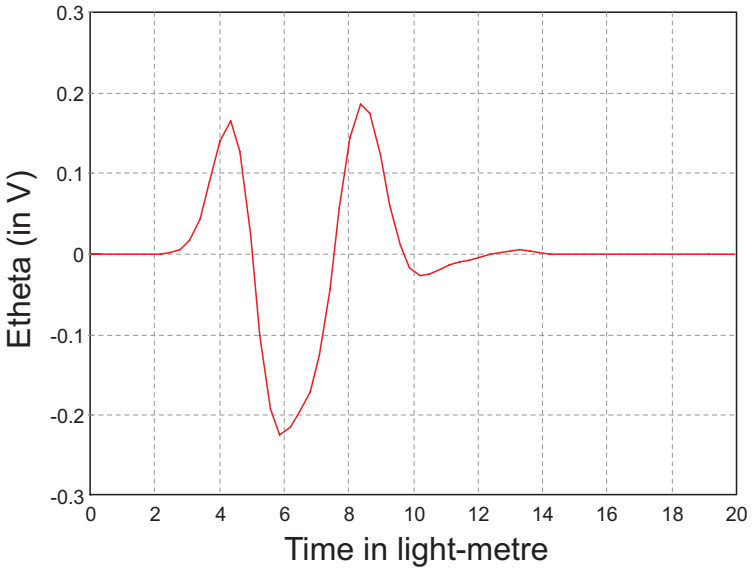


Figure 12-9: Response of the back scattered far field E_z of the cube.

13 The program ADAPTFEKO

13.1 Description

In examples with narrow resonances a fine frequency resolution is required to locate these resonances. If the frequency band is large, a very large number of analysis may be required if simple linear or multiplicative frequency stepping is used. ADAPTFEKO is used to overcome these problems. It uses an adaptive frequency sampling and interpolation, automatically using smaller steps near resonances and larger steps where the results are relatively smooth.

For each frequency it creates a `*.pre` file and calls PREFEKO and FEKO. The filenames are derived from the original name plus `_ada_` plus a numerical value (for example, the new files associated with `forked_dipole.pre` are `forked_dipole_ada_1.pre`, `forked_dipole_ada_2.pre`, ...).

13.2 Running ADAPTFEKO

ADAPTFEKO is started automatically by RUNFEKO if the FR card contains the flag for adaptive frequency sampling (see sections 8.2 and 10.2.29). The syntax is

```
runfeko filename
runfeko filename --adaptfeko-options options
```

where the optional argument `options` in the second line may be

`--keep-files`	All solution files (`*.pre`, `*.fek`, `*,out`, etc.) are preserved.
`--restart x`	Restart an adaptive frequency analysis using results for the frequency points `1`... `(x-1)` obtained in a previous run. (Then the previous run must have used `--keepfiles`.)

13.3 The *.pre input file

The `*.pre` file is created as for linear or multiplicative stepping. The only requirement is that only one FR card is used and that this card requests a continuous frequency band.

The variable `#adaptfreq` is defined automatically at the start of the single frequency input files and this variable may be used to allow, for example, adaptive meshing. One should not directly assign this variable inside the `*.pre` file as this will overwrite the value specified by ADAPTFEKO at the top of the file. If one needs this variable (for example to run PREFEKO during model setup when using adaptive meshing), one may use the DEFINED function

```
!!if (not(defined(#adaptfreq))) then
#adaptfreq = 250.0E6
!!endif
```

13.4 ADAPTFEKO example

As an example we will consider the input impedance of a simple forked dipole shown in figure 13-1. The input file `forked_dipole.pre` is as follows (this file is located in the `examples\utils\adaptfeko` subdirectory of the FEKO installation.)

```
** Forked dipole antenna
** Full MoM solution with adaptive frequency sampling

** Frequency (for the discretisation)
#lam = #c0 / 3.0e8

** Segmentation parameters
IP                              #lam/1000           #lam/20

** Define some points
DP   P1                        -0.01      0         0.5
DP   P2                         0         0         0.01
DP   P3                         0.01      0         0.466
DP   P4                         0         0        -0.01

** Half of the wire
LA   0
BL   P1   P2
BL   P2   P3

** Symmetry
SY   1    0    0    1

** The feed segment
LA   1
BL   P2   P4

** End of geometry
EG   1    0    0    0    0


** Excitation
A1   0    1                    1         0

** Adaptive frequency band
FR        2                    100.0e6             300.0e6
```

```
** Just compute the impedance
OS   0

** End
EN
```

Figure 13-1: Forked dipole used in the ADAPTFEKO example

Note that we do not use adaptive meshing as the model is quite small. This avoids the trouble associated with small discontinuities resulting from changes in the mesh.

Running ADAPTFEKO results in the file `forked_dipole.bof` which can be processed in POSTFEKO similar to any other BOF file. In addition, the *Use continuous frequency* field on all the result panels is enabled. This allows plotting the continuous result as a function of frequency or extracting the result at a specific frequency (see section 6.5.2).

For this example we create a new source data graph of *Solution* 1, *Source* 1 and plot the real and imaginary parts of the input impedance. The resulting input impedance is given in figure 13-2. Figure 13-3 shows results over a small frequency band where a solution obtained with linear stepping is added to the same graph. Note how close the results match even though ADAPTFEKO used only a single sample in this region.

This example was taken from the paper "Efficient wide–band evaluation of mobile communications antennas using [Z] or [Y] matrix interpolation with the method of moments," by K. L. Virga and Y. Rahmat-Samii, in the *IEEE Transactions on Antennas and Propagation*, vol. 47, pp. 65–76, January 1999. In that paper they considered the input admittance of a forked monopole. For comparison we have plotted the input admittance, scaled it to mSiemens and scaled the bottom axis to h/λ with $h = 1$m. Since we consider a dipole while their example uses a monopole we need to multiply the admittance with a further factor of 2. The result is shown in figure 13-4 and compares very well to the published result.

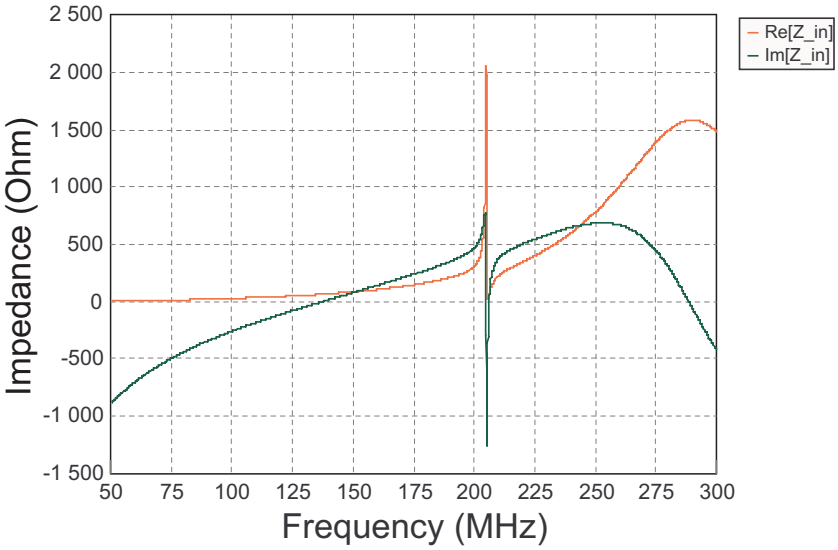


Figure 13-2: Input impedance of the forked dipole

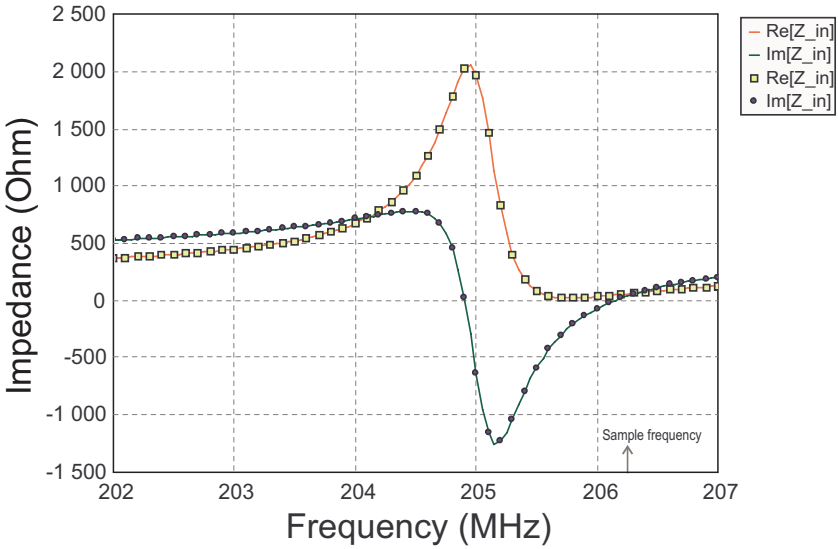


Figure 13-3: Input impedance of the forked dipole around the resonance point. The squares and circles represent values calculated at the discrete frequencies. The marker at 206.2 MHz denotes the only adaptive sample point in this band.

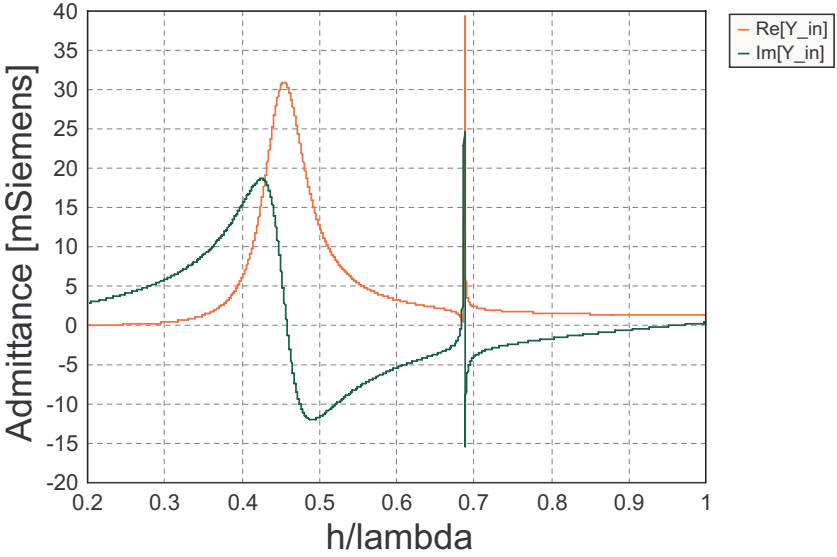


Figure 13-4: Input admittance of a forked monopole derived by multiplying the admittance of a forked dipole by a factor of 2.

14 Description of the output file of FEKO

The program FEKO writes all the results to an output file `*.out`. In this section the parts of the output file are described.

14.1 Geometric data

First the geometric data is given (if it has not been suppressed at the EG card). For the metallic triangles the following extract is written:

```
                    DATA OF THE METALLIC TRIANGLES

 no. Label      x1 in m     y1 in m     z1 in m       edges
     medium     x2 in m     y2 in m     z2 in m
                x3 in m     y3 in m     z3 in m       area in m*m
   1    0        1.2733       .0000       .0000        1     2
        0        1.9100       .0000       .0000
                 1.7646       .7309       .0000      2.3268E-01
   2    0        1.1027       .6367       .0000       -1     3     4
        0        1.2733       .0000       .0000
                 1.7646       .7309       .0000      2.1874E-01
```

In the first column the number of the triangle is written. In the second column the label is given, followed by the medium in which the triangle is situated. A 0 means that it is in free space. The next three columns are the x, y and z coordinates of the three corner points of the triangles. In the first row of each triangle follows another list of the numbers of the edges of the adjacent triangles. A positive sign indicates that the positive current direction is away from the triangle. A negative sign indicates that the positive current direction is towards the triangle. Below the edge numbers the area of the triangle is given in m^2.

Following this is an extract of the data for the edges between the triangle. Whenever two triangles have two common vertices, such an edge is generated.

```
                    DATA OF THE METALLIC EDGES (with MoM)

                                                 triangle no.    points of tr. ...
 no.  type      length/m        media      KORP   KORM   POIP   POIM
     1   1   5.3033E-01       0     -1      1      2      1      1
     2   1   3.7500E-01       0     -1      1     33      2      3
     3   1   3.7500E-01       0     -1      1     65      3      2
                                                  information on symmetry
                                               yz     xz     xy  status
                                                0     49     93  unknown
                                                0     -2     94        0
                                                0     50     -3  unknown
```

Each edge is assigned a consecutive number, which appears in the first column. The length of the edge is given in the second column and the medium in which the edge is found appears in the third column. On an edge there are exactly two triangles. In the columns KORP and KORM the numbers of these two triangles are given and the positive current direction is from the triangle KORP to the triangle KORM. In the column POIP the number of the corner point of the triangle KORP, which is opposite to the edge, is given. The same applies to the column POIM.

The next four columns contain information concerning the symmetry. In the column yz the number of the edge appears, corresponding to the plane $x = 0$ (yz plane) of symmetry. A positive sign indicates that the currents are symmetric and a negative sign indicates that the currents are anti-symmetric. If there is a 0 present in this column, then a symmetric edge does not exist. The same applies to the next columns xz and xy concerning the planes $y = 0$ and $z = 0$.

The last column with the heading STATUS has the following meaning: If unknown appears in it, the edge has an unknown status. The applicable coefficient of the current basis function cannot be determined from the symmetry, but has to be determined form the solution of the matrix equation. If 0 is present in the STATUS column, then the coefficient of the current basis function is 0 due to electric or magnetic symmetry and does not have to be determined.

If there is any other number in the STATUS column then this number indicates another edge whose coefficient is equal to (positive sign in the column STATUS) or the negative of (negative sign in the column STATUS) of the coefficient of the current basis functions. From symmetry the coefficient of the current triangle does not have to be determined.

The data of the dielectric triangles (surface current method) differ only slightly.

```
                    DATA OF THE DIELECTRIC TRIANGLES

 no.       label     x1 in m     y1 in m     z1 in m       edges
          medium     x2 in m     y2 in m     z2 in m
          medium     x3 in m     y3 in m     z3 in m
                       nx          ny          nz          area in m*m
        1      0  -1.0000E-01  2.0000E-01 -8.5000E-01          1       2       3
               1  -1.0000E-01  4.0000E-01 -8.5000E-01
               0   1.0000E-01  2.0000E-01 -8.5000E-01
                   0.0000E+00  0.0000E+00 -1.0000E+00       2.0000E-02
        2      0   1.0000E-01  4.0000E-01 -8.5000E-01          -1       4       5
               1   1.0000E-01  2.0000E-01 -8.5000E-01
               0  -1.0000E-01  4.0000E-01 -8.5000E-01
                   0.0000E+00  0.0000E+00 -1.0000E+00       2.0000E-02
```

In this case an additional line gives the components (nx, ny, nz) of the normal vector of each triangle.

For the edges the extract is

```
                    DATA OF THE DIELECTRIC EDGES (with MoM)

                                                  triangle no.   points of the triangle ...
 no.  type      length/m        media     KORP   KORM  POIP  POIM  POIA  POIE
    1   3    2.8284E-01        0      1      1      2     1     1     3     2
    2   3    2.0000E-01        0      1      1      3     2     3     1     3
    3   3    2.0000E-01        0      1      1     26     3     3     1     2
                        electr. info of symmetry        magnet. info of symmetry
                        yz     xz     xy  status       yz     xz     xy  status
                         0      0     42  unknown       0      0     42  unknown
                         0      0     43  unknown       0      0     43  unknown
                         0      0     44  unknown       0      0     44  unknown
```

In addition to the data that is given for the metallic triangles, the following columns are provided `POIA`, `POIE`, `KNP` and `KNM`. The column `POIA` contains the number of the corner point of the triangle in `KORP`, where the basis function for magnetic surface current begins and the column `POIM` contains the number of the end point of the triangle where the basis function ends. The sizes `KNP` and `KNM` are the lengths when the vertices are connected to the middle of the opposite edge in the triangles `KORP` and `KORM`. The symmetry information is shown for the basis functions of both the equivalent electric or magnetic current densities.

The data for the segments follows the data for the triangles.

```
                          DATA OF THE SEGMENTS

 No.       label     x1 in m     y1 in m     z1 in m      nodes
           medium    x2 in m     y2 in m     z2 in m    length in m  radius in m
       1       0   0.0000E+00  0.0000E+00 -2.7700E+00         1
               0   0.0000E+00  0.0000E+00 -2.1625E+00   6.0750E-01   2.7700E-02
       2       0   0.0000E+00  0.0000E+00 -2.1625E+00        -1        2
               0   0.0000E+00  0.0000E+00 -1.5550E+00   6.0750E-01   2.7700E-02
       3       0   0.0000E+00  0.0000E+00 -1.5550E+00        -2        3
               0   0.0000E+00  0.0000E+00 -9.4750E-01   6.0750E-01   2.7700E-02
```

Here each segment is assigned a consecutive number. In the second column the label of the segment appears and below it the number of the medium in which it finds itself. A zero (0) means free space (vacuum). Then the coordinates of the begin and end points of the segment follow. In the previous row the numbers of the nodes, that are adjacent, appear. A positive sign for the node number indicates that the positive current direction is defined away from the segment. When there is a negative number then the positive direction is towards the segment.In the next row the length of the segment appears, followed by the radius.

For the data of the nodes between the segments a data table is given.

DATA OF THE NODES BETWEEN THE SEGMENTS

	no. of segment		points of segm.		...
No.	ISEGP	ISEGM	KNOP	KNOM	
1	1	2	2	1	
2	2	3	2	1	
3	3	4	2	1	

info of symmetry			
yz	xz	xy	status
0	0	5	unknown
0	0	6	unknown
0	0	7	unknown

The consecutive numbers of nodes are given in the first column. Then the number ISEGP and ISEGM of the two connected segments follow. A positive current direction is defined from the segment ISEGP to the segment ISEGM. The column KNOP indicates whether the begin point(KNOP = 1) of the segment ISEGP connects to the node or if it is the end point(KNOP = 2). The following four columns contain the data about the symmetry and are the same as for the metallic triangles described above.

If there are any connections between triangles and segments, then the following data is given.

GEOMETRIC DATA OF CONNECTIONS SEGMENTS - TRIANGLES

	Data of triang.		data of segm.			info of symmetry			
no.	DRENUM	DREPOI	SEGNUM	SEGPOI	angle	yz	xz	xy	status
1			11	1	360.0000	0	0	0	unknown
	15	1			45.0000				
	33	1			45.0000				
	55	1			45.0000				

Each connection point is assigned a consecutive number which is given in the first column. The number of the triangle at the connection point is given in the column DRENUM with the connecting vertex (1 to 3) in the column DREPOI. Likewise the connecting segment's number is given in the column SEGNUM and the connecting end in the SEGPOI column. (If the begin point of the segment is connected, SEGPOI=1; else the end point is connected and SEGPOI=2.) The column angle gives the angle that is formed by the triangle at the connection point (in radians). The meaning of the symmetry information in the last four columns is the same as that of the metallic triangles given above.

If dielectric volume elements are used, then the following data block is given:

```
                    DATA OF THE DIELECTRIC CUBOIDS

   No.        x1 in m      y1 in m      z1 in m
   label      x2 in m      y2 in m      z2 in m
   medium     x3 in m      y3 in m      z3 in m
              x4 in m      y4 in m      z4 in m
      1    0.0000E+00   0.0000E+00   0.0000E+00
      0    3.3333E-01   0.0000E+00   0.0000E+00
   Cube    0.0000E+00   3.3333E-01   0.0000E+00
           0.0000E+00   0.0000E+00   3.3333E-01
      2    0.0000E+00   0.0000E+00   3.3333E-01
      0    3.3333E-01   0.0000E+00   3.3333E-01
   Cube    0.0000E+00   3.3333E-01   3.3333E-01
           0.0000E+00   0.0000E+00   6.6667E-01
      3    0.0000E+00   0.0000E+00   6.6667E-01
      0    3.3333E-01   0.0000E+00   6.6667E-01
```

Each cuboid is given a consecutive number. The x, y and z corner points coordinates are given in the first three columns. The first row is the reference point. The second row is the corner point to which from the reference point the first basis function is defined. Further, the third and fourth rows define the next two basis functions with respect to the reference point.

In each dielectric cuboid there are three basis functions, in each coordinate direction one. The data of these basis functions is given in the following format:

```
          DATA OF THE BASIS FUNCTIONS FOR DIELECTRIC CUBOIDS

                                Symmetry information
 No. cuboidno. direc.    yz       xz       xy  status
     1        1  1         28       55      109  unknown
     2        2  1         29       56      110  unknown
     3        3  1         30       57      111  unknown
     4        4  1         31       58      112  unknown
     5        5  1         32       59      113  unknown
```

In the first column the consecutive number of the basis function is given. The next column indicates the number of the cuboid. The column direction indicates the direction of the basis function in the respective cuboid. 1 indicates that e.g. the basis function is defined from the reference point to the second corner point. The last four columns contain information concerning the symmetry properties of the cuboid, where the structure and the meaning is the same as with the other basis functions.

For the FEM, the data of the tetrahedral volume elements is printed in a table like this:

```
                    DATA OF THE TETRAHEDRAL VOLUME ELEMENTS (FEM)
```

```
no.       label     x1 in m     y1 in m     z1 in m  nodes
         medium     x2 in m     y2 in m     z2 in m  faces
                    x3 in m     y3 in m     z3 in m  edges
                    x4 in m     y4 in m     z4 in m  volume in m*m*m
       1 Union1  -2.3572E-02 -1.0940E-02 -1.4989E-02     1    2    3    4
            air  -1.3328E-02 -6.4075E-03 -2.6102E-02     1    2    3    4
                 -1.1854E-02 -1.4150E-02 -1.6861E-02     1    2    3    4    5    6
                 -7.2568E-03 -2.5013E-02 -1.4889E-02    1.3899E-07
       2 Union1  -7.2568E-03 -2.5013E-02 -1.4889E-02     4    5    6    7
            air  -9.6795E-03 -1.8920E-02 -1.3166E-02     5    6    7    8
                 -1.2531E-02 -1.6999E-02 -1.3380E-02     7    8    9   10   11   12
                 -8.8380E-03 -1.6914E-02 -1.6150E-02    8.7909E-09
       3 Union1  -7.2568E-03 -2.5013E-02 -1.4889E-02     4    5    6    8
            air  -9.6795E-03 -1.8920E-02 -1.3166E-02     5    9   10   11
                 -1.2531E-02 -1.6999E-02 -1.3380E-02     7    8   13   10   14   15
                 -1.2033E-02 -1.9460E-02 -1.0077E-02    8.8478E-09
```

With the provided header, this information should be self explanatory.

Thereafter information follows regarding the number of basis functions.

```
                        DATA FOR MEMORY USAGE

 Number of metallic triangles:          0       max. triangles:MAXNDR   =    62
 Number of dielectric triangles:        0
 Number of FEM surface triangles:      62
 Number of metallic segments:          37       max. segments: MAXNSEG  =    37
 Number dielectr./magnet. cuboids:      0       max. cuboids:  MAXNQUA  =     0
 Number of tetrahedral elements:     5177       max. tetrah.:  MAXNTETRA=  5177
 Number of edges in PO region:          0       max. edges:    MAXPOKA  =     0
 Number of wedges in PO region:         0       max. wedges:   MAXPOKL  =     0
 Number of Fock regions:                0       max. Fock reg.:MAXFOGE  =     0
 Number of polygonal surfaces:          0       max. surfaces: MAXPOLYF =     0
                                                max. corner p.:MAXPOLYP =     0
 Number of UTD cylinders:               0

 Number of metallic edges (MoM):     0  unknown:  0  max. edges:   MAXNKA    = 93
 Number of metallic edges (PO):      0  unknown:  0  (electr.) 0 (magnet.)
 Number of dielectric edges (MoM):   0  unknown:  0  (electr.)
                                     0             0  (magnet.)
 Number of dielectric edges (PO):    0  unknown:  0  (electr.) 0 (magnet.)
 Number of edges FEM/MoM surface:   93  unknown: 93  (electr.)
                                    93            93  (magnet.)
 Number of nodes between segments: 36  unknown: 36  max. nodes:   MAXNKNO   = 66
 Number of connection points:        0  unknown:  0  max. conn.:   MAXNV     =  0
 Number of dielectric cuboids:       0  unknown:  0  max. cuboids: MAXNQUA   =  0
 Number of magnetic cuboids:         0  unknown:  0
```

```
Number of basis funct. for MoM:   36  unknown: 36  max. basisf.  MAXNZEILE = 86
Number of basis funct. for PO:     0  unknown:  0  max. basisf.  MAXNKAPO  =  0

Memory requirement for MoM matrix: 144 rows * 222 columns = 31968 complex numbers
For the matrix, a memory of 31968 complex numbers is available (i.e. 499.5 kByte)
```

Here the data, e.g. the number of basis functions on the nodes between segments, can be extracted. It is also indicated how many have the status `unknown`, i.e. how many have to be determined by solving the matrix equation.

14.2 Excitation

The data here is structured depending on the means of excitation. For a voltage source on a segment the following data block is generated:

```
                EXCITATION BY VOLTAGE SOURCE AT SEGMENT

Number of voltage source:    N =   1
Frequency in Hz:             FREQ =  7.50000E+07
Wavelength in m:             LAMBDA =  3.99723E+00
Open circuit voltage in V:   |U0| =  1.00000E+00
Phase in deg.:               ARG(U0) =     0.00
Source at segment w. label:  ULA =     1
Absolute number of segment:  UNR =    11
```

If an incident plane wave is used then the output file has the following format:

```
      EXCITATION BY PLANE LINEAR POLARISED ELECTROMAGNETIC WAVE

Number of excitation:   N =   1
Frequency in Hz:        FREQ = 1.9467E+07
Wavelength in m:        LAMBDA = 1.5400E+01
Direction of incidence: THETA = -180.00  PHI =     .00
Dir. of polarisation:   ETA =    0.00
Direction of propag.:   BETA0X = -1.6736E-10
                        BETA0Y =  0.0000E+00
                        BETA0Z =  4.0800E-01
Field strength in V/m:  |E0X| =  1.00000E+00   ARG(E0X) =     0.00
(Phase in deg.)         |E0Y| =  0.00000E+00   ARG(E0Y) =     0.00
                        |E0Z| =   4.1021E-10   ARG(E0Z) =     0.00
```

The vector $\vec{\beta}$, whose components are given, is the vector which points in the direction of propagation. The vector $\vec{E}_0$ represents the direction of the electric field.

14.3 Currents and charges

The OS card can request the current distribution. The following data is given for each triangle

```
                VALUES OF THE CURRENT DENSITY VECTOR ON TRIANGLES in A/m

Triangle     centre                  JX              JY              JZ             ...
number    x/m    y/m    z/m      magn.     phase     magn.    phase    magn.     phase
 1      -.944   .000   .056   1.644E-03   19.10  0.000E+00    .00  3.716E-02 162.73
 2      -.889   .000   .111   1.184E-03  163.01  0.000E+00    .00  3.238E-02 157.26
 3      -.944   .000   .222   4.709E-03   12.49  0.000E+00    .00  2.784E-02 149.11
 4      -.889   .000   .278   2.032E-03 -170.73  0.000E+00    .00  2.081E-02 119.69
 5      -.944   .000   .389   4.285E-03   13.23  0.000E+00    .00  2.083E-02 100.40

                                                ...    Current magnitude in the
                                                            3 corner points
                                                     3.850E-02 3.882E-02 3.457E-02
                                                     3.145E-02 3.169E-02 3.445E-02
                                                     3.143E-02 3.446E-02 2.197E-02
                                                     2.109E-02 2.234E-02 2.146E-02
                                                     2.106E-02 2.467E-02 2.291E-02
```

At the position (x, y, z) the current density vector $\vec{J}$ in the complex form is given. The last three columns indicate the value for the surface current density in the three vertices of the triangles, where the value is the average of the current at the vertices of all three adjacent triangles. If the current is requested, the charge on each triangle is also written to the output file. Only the charge is given as the position of each triangle is the same as written for the currents.

```
            VALUES OF THE SURFACE CHARGE DENSITY ON TRIANGLES in As/m^2

Triangle          SIGMA
number         magn.      phase
        1   2.50469E-13   56.08
        2   3.55072E-13   42.60
        3   9.33040E-13   54.20
```

The current on the segments is written as

```
                    VALUES OF THE CURRENT IN THE SEGMENTS in A

Segment        centre                                        IX          ...
number           x/m          y/m          z/m        magn.     phase
        1   2.25000E+00  0.00000E+00  6.67500E-01  0.000E+00     0.00
```

```
       2   2.25000E+00  0.00000E+00  5.02500E-01 0.000E+00    0.00
       3   2.25000E+00  0.00000E+00  3.37500E-01 0.000E+00    0.00
                                    IY                IZ
                                 magn.    phase    magn.    phase
                              0.000E+00    0.00 2.208E-03  145.51
                              0.000E+00    0.00 6.118E-03  146.21
                              0.000E+00    0.00 9.000E-03  147.38
```

With the associated charge

```
          VALUES OF THE LINE CHARGE DENSITY ON SEGMENTS in As/m

Segment               Q
number          magn.       phase
       1    4.26233E-11   55.51
       2    3.28910E-11   58.02
       3    2.28537E-11   62.48
```

For every voltage source the current at the feed point is determined and thus the impedance. The following is the result:

```
                          DATA OF THE VOLTAGE SOURCE NO.   1

                     real part imag. part      magn.          phase
 Current    in A    1.0888E-02 -4.4405E-03  1.1759E-02       -22.19
 Admitt.    in A/V  1.0888E-02 -4.4405E-03  1.1759E-02       -22.19
 Impedance in Ohm   7.8747E+01  3.2116E+01  8.5044E+01        22.19

                                Power in Watt: 5.44395E-03
```

14.4 Finite conductivity

Firstly the block with the set of characteristics for the single labels is displayed:

```
                            DATA OF LABELS

Label     2: DOSKIN =  3             DOLAST =  0                  ...
          Triangle thickness:  5.00000E-03 m
          Sigma =   1.000E+05 S/m   Mue_r =   1.000E+00
          Penetration depth of the skin effect:  1.59210E-04 m

All segments and triangles without a listed label are perfectly
                              DOCOVR =  0

                           tan(delta_mu) =   0.000E+00

                                conducting
```

After the calculation of the currents the losses that result from finite conductivity are displayed.

```
                              POWER LOSS (in Watts)

     |              ...in the segments                 |   ...in the triangles
Label| skineffect  ohm.loss    distr.load   coating     | skineffect  ohm. loss
    2| 0.0000E+00  0.0000E+00  0.0000E+00   0.0000E+00  | 5.4620E-07  1.7386E-08
total| 0.0000E+00  0.0000E+00  0.0000E+00   0.0000E+00  | 5.4620E-07  1.7386E-08

          Total loss in the segments:          0.0000E+00 W
          Total loss in the triangl.:          5.6359E-07 W
          Loss (total)         :               5.6359E-07 W
          Efficiency of the ant.  :               99.9877 %
```

In the first column the label is displayed, the lowest row displays the sum.

14.5 Near fields

If the near field is calculated, the following data is displayed:

```
                        VALUES OF THE ELECTRIC FIELD STRENGTH in V/m

                           (total field, incident and scattered)

                 LOCATION                              EX                  EY    ...
medium    X/m          Y/m          Z/m          magn.    phase      magn.    phase
   0  0.00000E+00 0.00000E+00 -1.00000E+00   6.70088E+01 99.86  7.65636E-01 166.42
   0  1.00000E-01 0.00000E+00 -1.00000E+00   6.46235E+01 74.23  1.14589E+00 166.13
   0  2.00000E-01 0.00000E+00 -1.00000E+00   6.23014E+01 47.98  1.55289E+00 165.83
   0  3.00000E-01 0.00000E+00 -1.00000E+00   5.99908E+01 21.51  1.95743E+00 163.95

                                                    ...              EZ
                                                                magn.    phase
                                                          6.89061E+01 126.74
                                                          7.17685E+01  98.76
                                                          7.35678E+01  70.70
                                                          7.41473E+01  42.30
```

Displayed are the position as well as the individual components of the electric and the magnetic field strength. This is — if not requested otherwise at the FE card — the total value of the field, i.e. the sum of the incident wave and the scattered field.

If the electric field inside dielectric cuboids is determined, then the value for the SAR (*specific absorption rate*) and the cuboid number are also given:

VALUES OF THE ELECTRIC FIELD STRENGTH in V/m

inside the dielectric cuboids

LOCATION			EX		EY		EZ		...
X/m	Y/m	Z/m	magn.	phase	magn.	phase	magn.	phase	
0.050	0.050	0.050	5.776E+00	59.89	1.259E+01	-177.82	1.415E+02	-125.12	
0.050	0.050	0.150	2.192E+01	33.75	4.114E+00	122.93	1.640E+02	-130.45	
0.050	0.050	0.250	2.584E+01	31.18	3.420E+00	19.21	1.679E+02	-137.51	
0.050	0.050	0.350	2.625E+01	22.29	8.499E+00	-24.72	1.551E+02	-144.87	

...	SAR in W/kg	cuboid no.
	0.000E+00	1
	0.000E+00	2
	0.000E+00	3
	0.000E+00	4

14.6 Far fields

If the far field is calculated, the following block in this form is displayed:

VALUES OF THE SCATTERED ELECTRIC FIELD STRENGTH IN THE FAR FIELD in V
Factor e^(-j*BETA*R)/R not considered

LOCATION		ETHETA		EPHI		directivity in dB			...
THETA	PHI	magn.	phase	magn.	phase	vert.	horiz.	total	
90.00	0.00	1.235E+00	168.98	0.000E+00	0.00	7.1722	-999.9999	7.1722	
90.00	2.00	1.233E+00	168.90	0.000E+00	0.00	7.1583	-999.9999	7.1583	
90.00	4.00	1.227E+00	168.65	0.000E+00	0.00	7.1166	-999.9999	7.1166	

POLARISATION		
axial r.	angle	direction
0.0000	180.00	LINEAR
0.0000	180.00	LINEAR
0.0000	180.00	LINEAR

Gain is a factor of 1.00000E+00 (0.00 dB) larger than directivity

The directivity/gain is based on an active power of 4.88015E-03 W
and on a power loss of 0.00000E+00 W

For incident plane waves, the values that are displayed here are the values of the scattered field, i.e. the incident field is not taken into account. However, for any other sources (e.g. elementary Hertzian dipoles or impressed radiation pattern or transmission line), the fields radiated by the excitation are considered.

In the far-field a complex field strength $\vec{E}_{\text{far}}$ is defined using the relation

$$\lim_{R\to\infty} \vec{E}(\vec{r}) \;=\; \frac{e^{-j\beta_0 R}}{R}\,\vec{E}_{\text{far}}$$

with a large distance $R = |\vec{r}|$ which tends to infinity (and which in the FEKO calculations is identical to infinity). Please note that the dimension of $\vec{E}_{\text{far}}$ is voltage due to this extra distance factor R.

In the `*.out` file the ϑ (vertical) and φ (horizontal) components of $\vec{E}_{\text{far}}$ are tabulated by magnitude and phase, i.e. $\vec{E}_{\text{far},\vartheta}$ and $\vec{E}_{\text{far},\varphi}$. With POSTFEKO also results for other polarisations can be extracted. The corresponding formulas used are then

$$\vec{E}_{\text{far,S}} \;=\; \frac{1}{\sqrt{2}}\left(\vec{E}_{\text{far},\varphi} - \vec{E}_{\text{far},\vartheta}\right)$$

for S-polarisation,

$$\vec{E}_{\text{far,Z}} \;=\; \frac{1}{\sqrt{2}}\left(\vec{E}_{\text{far},\varphi} + \vec{E}_{\text{far},\vartheta}\right)$$

for Z-polarisation,

$$\vec{E}_{\text{far,LHC}} \;=\; \frac{1}{\sqrt{2}}\left(\vec{E}_{\text{far},\varphi} + j\vec{E}_{\text{far},\vartheta}\right)$$

for left-hand cirular polarisation, and

$$\vec{E}_{\text{far,RHC}} \;=\; \frac{1}{\sqrt{2}}\left(\vec{E}_{\text{far},\varphi} - j\vec{E}_{\text{far},\vartheta}\right)$$

for right-hand circular polarisation.

If the excitation is an incident wave, the results include the radar cross section. In the case of voltage sources, the gain or directivity are included (see the parameters of the FF card).

For the radar cross section, the incident plane wave with complex amplitude $\vec{E}_0$ carries a power density of

$$S_i \;=\; \frac{1}{2}\cdot\frac{|\vec{E}_0|^2}{Z_{F0}}$$

(Z_{F0} denotes the wave impedance of the surrounding medium) which gets scattered on the object and a wave is reflected with the scattered power density

$$S_s \;=\; \frac{1}{2}\cdot\frac{|E_{\text{far},\vartheta}|^2 + |E_{\text{far},\varphi}|^2}{Z_{F0}} \quad .$$

The radar cross section (RCS) σ is then defined as

$$\sigma \;=\; \lim_{R\to\infty}\; 4\pi R^2\,\frac{S_s}{S_i} \;=\; \lim_{R\to\infty}\; 4\pi\,\frac{|R\,E_{\text{far},\vartheta}|^2 + |R\,E_{\text{far},\varphi}|^2}{|\vec{E}_0|^2} \quad .$$

For antenna and general radiation problems, as mentioned above, FEKO is computing either the gain or the directivity depending on the FF card setting (this applies to the values tabulated in the `*.out` file only, in POSTFEKO any quantity can be selected). Let us assume that P_t is the source power and P_v are some losses in the structure (e.g. dielectric losses), then a power $P_r = P_t - P_v$ will be radiated. The directivity (as compared to an isotropic point source) is then defined as

$$D \;=\; 4\pi\, R^2\, \frac{S_s}{P_r} \;=\; \frac{2\pi}{Z_{F0}} \cdot \frac{|R\, E_{\mathrm{far},\vartheta}|^2 + |R\, E_{\mathrm{far},\varphi}|^2}{P_r} \quad .$$

For the gain a similar definition is used, just now the source power P_t and not the radiated power P_r is acting as reference:

$$G \;=\; 4\pi\, R^2\, \frac{S_s}{P_t} \;=\; \frac{2\pi}{Z_{F0}} \cdot \frac{|R\, E_{\mathrm{far},\vartheta}|^2 + |R\, E_{\mathrm{far},\varphi}|^2}{P_t} \quad .$$

Between gain and directivity there is the relation

$$\frac{G}{D} \;=\; \frac{P_r}{P_t} \;=\; \frac{P_t - P_v}{P_t} \;=\; \eta, \tag{14-1}$$

where η represents the antenna efficiency.

The last three columns of the far-field ouput give the polarisation information of the scattered wave. In general the polarisation is elliptical as shown in figure 14-1. The coordinates are $\vec{e}_r$, $\vec{e}_\vartheta$ and $\vec{e}_\varphi$, and the view is in the direction of the propagation of the wave ($\vec{e}_r$).

In order to evaluate these quantities, let us define the magnitude and phase of the far-field components as

$$E_{\mathrm{far},\vartheta} \;=\; A \cdot e^{j\alpha} \qquad E_{\mathrm{far},\varphi} \;=\; B \cdot e^{j\beta} \quad .$$

Using the abbreviation $\tau = \omega t - \beta_0 r$ one finds the temporal field strenght vector in space as

$$\vec{E}(\tau) \;=\; \frac{A}{r} \cdot \cos(\tau + \alpha) \cdot \vec{e}_\vartheta + \frac{B}{r} \cdot \cos(\tau + \beta) \cdot \vec{e}_\varphi \quad .$$

This equation describes then the polarisation ellipse of figure 14-1. One can find the minimum and maximum values of the field strength magnitude at these times

$$\tau_1 \;=\; -\frac{1}{2} \cdot \arctan \frac{A^2 \cdot \sin(2\alpha) + B^2 \cdot \sin(2\beta)}{A^2 \cdot \cos(2\alpha) + B^2 \cdot \cos(2\beta)}$$

and

$$\tau_2 \;=\; \tau_1 + \frac{\pi}{2} \quad .$$

Let $E_1 = |\vec{E}(\tau_1)|$ and $E_2 = |\vec{E}(\tau_2)|$ and assuming that we have $E_1 > E_2$, then according to figure 14-1 we have $E_{max} = E_1$ und $E_{min} = E_2$. The axial ratio is then defined as

$$v \;=\; \frac{E_{min}}{E_{max}} = \frac{E_2}{E_1}$$

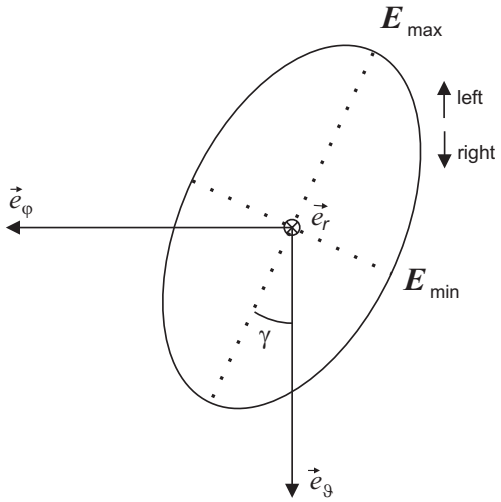


Figure 14-1: Elliptic polarisation in the far field

A ratio of 0 means that the wave is a linearly polarised wave, but if the ratio has a value of 1 then it is a circularly polarised wave. The direction of rotation is right hand circular (RHC) for $0 < \alpha - \beta < \pi$ and left hand circular (LHC) for $\pi < \alpha - \beta < 2\pi$.

FEKO also computes and prints the polarisation angle γ. It is the angle between the major axis of the polarisation ellipse and the unit vector $\vec{e}_\vartheta$ and can be computed using

$$\gamma = \arctan\frac{B \cdot \cos(\tau_1 + \beta)}{A \cdot \cos(\tau_1 + \alpha)}$$

If the FF card is used with NTHETA ≥ 2 and NPHI ≥ 2 the Poynting vector is integrated over the specified sector, see the detailed discussion given with the FF card in section 10.2.28. The result is the radiated power and is given below the field values.

When analysing an antenna the source power (calculated from the input impedance) should equal the integral of the radiated power over the surface of a sphere. This may be used as a partial validation of the result. Note that power losses in dielectrics and finite conductivity should be taken into account separately.

The use may also elect to integrate the far field power without writing the field values to the output file (using the FF card with FFREQ = 3). FEKO then produces the output

```
VALUES OF THE SCATTERED ELECTRIC FIELD STRENGTH IN THE FAR FIELD in V
Factor e^(-j*BETA*R)/R not considered

Integration of the normal component of the Poynting vector in the angular
```

```
grid DTHETA =    5.00 deg. and DPHI =    5.00 deg. (    2701 sample points)
  angular range THETA        angular range PHI        radiated power
 -2.50 .. 182.50 deg.      -2.50 .. 362.50 deg.     1.05642E-03 Watt
  0.00 .. 180.00 deg.       0.00 .. 360.00 deg.     1.04095E-03 Watt
Polarisation dependent radiated power:
     horizontal polarisation:    3.63158E-04 Watt ( 34.89 %)
     vertical polarisation:      6.77792E-04 Watt ( 65.11 %)
     S polarisation:             5.19105E-04 Watt ( 49.87 %)
     Z polarisation:             5.21845E-04 Watt ( 50.13 %)
     left hand circular pol.:    7.13564E-04 Watt ( 68.55 %)
     right hand circular pol.:   3.27386E-04 Watt ( 31.45 %)
```

where the first line of total power is calculated assuming that each specified point lies at the centre of an incremental integration area. The effective area is therefore slightly larger than the area defined by the start and end angles. The second line gives the power integrated over an area defined by the start and end angles. For example, one may request an integration from $\varphi = 0°$ to $\varphi = 350°$ and $\vartheta = 5°$ to $\vartheta = 175°$ both in $10°$ increments in which case the first total will give the total power through the sphere. One may also request the integration from $\varphi = 0°$ to $\varphi = 360°$ and $\vartheta = 0°$ to $\vartheta = 180°$ in which case the second total will give the correct total power through the sphere.

The polarisation dependent power on the second block is calculated according to the effective area of the second line. The definitions of S and Z polarisation can be found in section 6.7.

14.7 S-parameters

If S-parameters have been requested with an SP card, FEKO prints different tables to the output file. The first lists the impedance at each port (all sources that are active when the SP card is processed are considered as ports).

LOAD IMPEDANCES AT PORTS

port	impedance in Ohm
1	5.00000E+01
2	1.00000E+02
3	5.00000E+01

Then the S-parameters are listed for each source as shown below. Note that sources whose amplitude are set to exactly zero are only used as sink ports, i.e. they are not excited and no such block is created. All the ports are loaded and FEKO therefore also writes this information to the output file. The second data line below gives S_{21} or the coupling to port 2 when port 1 is excited. In the second block here under the first line gives S_{13} or the coupling into port 1 when port 3 is excited.

SCATTERING PARAMETERS

	ports				magnitude		phase
	sink	source	real part	imag. part	linear	in dB	in deg.
S	1	1	6.14622E-02	-3.53596E-01	3.58898E-01	-8.90	-80.14
S	2	1	3.61992E-03	5.42992E-03	6.52594E-03	-43.71	56.31
S	3	1	-1.46490E-03	1.73598E-02	1.74215E-02	-35.18	94.82

...

SCATTERING PARAMETERS

	ports				magnitude		phase
	sink	source	real part	imag. part	linear	in dB	in deg.
S	1	3	-1.31791E-03	1.74114E-02	1.74612E-02	-35.16	94.33
S	2	3	9.17744E-01	1.08299E-01	9.24112E-01	-0.69	6.73
S	3	3	3.49405E-01	-4.49374E-02	3.52282E-01	-9.06	-7.33

14.8 Computation time and peak memory

The final section in the output file gives an overview of the computation, in seconds, time in a tabular format:

SUMMARY OF REQUIRED TIMES IN SECONDS

	CPU-time	runtime
Reading and constructing the geometry	0.031	0.031
Checking the geometry	0.000	0.000
Initialisation of the Greens function	0.000	0.000
Calcul. of coupling for PO/Fock	0.000	0.000
Calcul. of matrix A	0.031	0.031
Calcul. of vector Y (right side)	0.000	0.000
Solution of the linear set of eqns.	0.016	0.016
Determination of surface currents	0.000	0.000
Calcul. of losses	0.000	0.000
Calcul. of electric near field	0.000	0.000
Calcul. of magnetic near field	0.000	0.000
Calcul. of far field	0.031	0.031
other	0.000	0.000
total times:	0.109	0.109

This table is followed by an output of the peak memory (main memory, excluding possible out-of-core files) usage which FEKO encountered during any solution phase:

```
Peak memory usage during the whole solution:   41.275 MByte
```

15 Copyright notices and acknowledgements

The copyright for FEKO lies with

Copyright 1998 – 2005: **EM Software & Systems-S.A. (Pty) Ltd**

However, FEKO uses also third-party libraries and components (both commercial and freely available ones). For some of the libraries the usage condition is that the corresponding copyright notices must be reproduced in the accompanying documentation. This shall be done here.

15.1 Copyright to Voronoi

The preprocessor PREFEKO uses part of the program `voronoi` to execute a Delaunay triangulation for some of the geometric cards. The copyright declaration for `voronoi` is as follows:


```
The author of this software is Steven Fortune.  Copyright (c) 1994 by
AT&T Bell Laboratories.
Permission to use, copy, modify, and distribute this software for any
purpose without fee is hereby granted, provided that this entire notice
is included in all copies of any software which is or includes a copy
or modification of this software and in all copies of the supporting
documentation for such software.
THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED
WARRANTY.  IN PARTICULAR, NEITHER THE AUTHORS NOR AT&T MAKE ANY
REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY
OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.
```


15.2 Copyright to SuperLU

The FEKO kernel is using for some parts the SuperLU library. The copyright declaration for `SuperLU` is as follows:


```
Copyright (c) 2003, The Regents of the University of California, through
Lawrence Berkeley National Laboratory (subject to receipt of any required
approvals from U.S. Dept. of Energy)

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

(1) Redistributions of source code must retain the above copyright notice,
```

this list of conditions and the following disclaimer.
(2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
(3) Neither the name of Lawrence Berkeley National Laboratory, U.S. Dept. of Energy nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

15.3 Copyright of libcurl

EDITFEKO makes use of `libcurl` to connect to the EMSS web-site to check for updates. The copyright declaration for `libcurl` is as follows:

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2005, Daniel Stenberg, <daniel@haxx.se>.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

15.4 Qwt project usage

For CADFEKO the figures in the mesh information dialogs are based, in part, on the work of the Qwt project (see `http://qwt.sf.net`).

15.5 HOOPS and Parasolid

POSTFEKO and CADFEKO use the HOOPS® 3D Application Framework from Tech Soft America for the 3D display and rendering. CADFEKO also uses the Parasolid® Kernel Modeller from UGS to represent the geometry.

15.6 MeshSim

CADFEKO uses MeshSim™ from Simmetrix Inc. to mesh the geometry.

Index

** card, 9-3, 10-5
2D results, 6-10
3D results, 6-7
3D view, 4-4

A0 card, 10-9
A1 card, 10-12
A2 card, 10-13
A3 card, 10-14
A4 card, 10-16
A5 card, 10-18
A6 card, 10-19
A7 card, 10-21
AC card, 10-23
ADAPTFEKO, 13-1
adaptive
    frequency sampling, 6-10, 6-11, 13-1
add series, 6-12
advanced solution options, 4-46
AE card, 10-26
AI card, 10-29
angles
    Kardan, 9-92
anisotropic layers, 10-111
ANSYS, 9-50
AP card, 10-31
aperture, 10-31
AR card, 10-38
array sizes, 2-16, 2-17
AS card, 10-43
assemblies, 4-34
AutoCAD file, 9-42
AV card, 10-49
AW card, 10-51
axes
    in POSTFEKO, 6-6

Beziér, 4-21
BL card, 9-4
BO card, 10-54
Boolean operations, 4-26
border
    PO correction edge, 9-53
    PO correction wedge, 9-58
BP card, 9-6
BQ card, 9-8
BT card, 9-10

CableMod, 10-23, 10-62
CADFEKO, 4-1
    availability, 4-1
    overview, 4-3
calculations
    on series, 6-12
    variables, 4-15
capacitance
    loading, 10-94, 10-97, 10-98
card editing, 5-6
CB card, 9-12
CDB file, 9-50
CF card, 10-56
CFIE, 10-56
CFM files, 4-5, 4-46
CFX files, 4-5
CG card, 10-58
check for updates, 5-5
circular arcs, 4-21
circular cone, 9-54
circular dependencies, 4-13
circular disc, 9-59
circular hole, 9-72
CL card, 9-14
clash, 4-33
clear model, 4-5
click mouse, 4-5
CM card, 10-62
CN card, 9-16
CO card, 10-63
coarse segmentation, 9-52
coating of wires, 10-63
coaxial attachment approximation, 10-16
coil, 9-31
collapse tree items, 4-11
colour, 4-6
    dielectrics, 4-31
colour by label, 6-5
colour by medium, 6-5
command line parameters, 4-46, 6-1

comments, 4-48, 9-3, 10-5
complexity, reducing, 4-29
CONCEPT file, 9-47
cones, 4-18, 9-54
connection points
    definition, 2-2
constructing solids, 4-31
contact us, 1-3
continuous frequency, 6-10, 6-11
control cards, 2-1, 10-1
    **, 10-5
    A0, 10-9
    A1, 10-12
    A2, 10-13
    A3, 10-14
    A4, 10-16
    A5, 10-18
    A6, 10-19
    A7, 10-21
    AC, 10-23
    AE, 10-26
    AI, 10-29
    AP, 10-31
    AR, 10-38
    AS, 10-43
    AV, 10-49
    AW, 10-51
    BO, 10-54
    CF, 10-56
    CG, 10-58
    CM, 10-62
    CO, 10-63
    DA, 10-66
    DI, 10-69
    EN, 10-70
    FE, 10-71
    FF, 10-79
    FR, 10-83
    GF, 10-86
    L4, 10-92
    LD, 10-94
    LE, 10-95
    LP, 10-97
    LS, 10-98
    LZ, 10-99
    OF, 10-100
    OS, 10-101
    PS, 10-103
    PW, 10-105
    SK, 10-111
    SP, 10-117
    TL, 10-118
control panel
    POSTFEKO, 6-5
copy, 4-24, 4-30
    geometry, 9-91
    multiple, 4-24
copyright, 15-1
coupling, 10-117
    transmission line, 10-23, 10-62
create
    geometry, 4-16
    solid regions, 4-31
cuboidal volume elements, 9-17, 9-20, 9-82
cuboids, 4-18, 9-17, 9-81, 9-82
    definition, 2-3
current sources
    line segment, 10-23, 10-29, 10-49
currents
    calculation request, 10-101
    display, 6-8
curves, 4-31
    arc, 9-14
    creating surfaces, 4-26
cutplanes, 4-10, 6-7
    and selection, 4-35
    through solids, 4-10
cylinders, 4-18, 9-105
    dielectric, 9-20
    UTD region, 9-100

DA card, 10-66
database browser, 6-17
delete
    elements, 4-41
    faces and edges, 4-29
    series, 6-12
details tree, 4-11
DI card, 10-69
dielectric, 2-14, 9-65, 10-69
    colour, 4-31
    cuboid cylinder, 9-20
    cuboids, 9-17, 9-81, 9-82
    for mesh elements, 4-45

medium, 4-31, 4-47
sphere, 10-86
thin sheet, 10-111
diffraction theory, 9-98
dimension, scaling, 9-88
dipole array aperture, 10-31
direction
normal vectors, 4-6, 4-20, 4-27, 4-29
triangles, 4-44
disc, 9-59
discrete elements, 10-97–10-99
display
items, 4-10
settings, 4-1
distorted elements, 4-43
distributed load, 10-94
DK card, 9-17
DP card, 7-10, 9-19
draggin mouse, 6-4, 6-14
dragging mouse, 4-5
duplicate elements, removing, 4-42
DXF file, 9-42
dynamic memory management, 2-16
DZ card, 9-20

edges, 4-11
definition, 2-2
delete, 4-29
length, 4-36, 4-40
properties, 4-38
show selection, 4-36
edit geometry, 4-23
EDITFEKO, 5-1
card editor, 5-6
edit menu, 5-3
file menu, 5-3
keystrokes, 5-10
OPTFEKO mode, 5-9
options, 5-5
preferences, 5-5
superuser mode, 5-6
variables, 5-9
editor, 4-48
efficiency, 10-105
EFIE, 10-56
EG card, 9-22
EL card, 9-25
electric fields
calculating, 10-71
elements, 9-10
count, 4-40
creation, *see* geometry cards
deleting, 4-41
distorted, 4-43
free edges, 4-43
searching, 6-7
ellipse, 4-19
ellipsoid, 9-25
elliptical arcs, 4-21
elliptical hole, 9-72
ELSE statement, 7-9
EN card, 10-70
end of geometry, 9-22
end of input file, 10-70
environment variables, 2-22
EPSENT, 9-23
equivalent aperture, 10-31
excitation, *see* sources
EXIT command, 7-11
expand tree items, 4-11
explode, 4-31
export, 4-5, 4-46
geometry, 4-21
images, 6-16
Parasolid, 4-16, 4-21
raw data, 6-15
expressions, 4-13
extents, 4-16
extrude parts, 4-26

faces, *see* surfaces, 4-11, 4-19, 4-31
delete, 4-29
metal, 4-33
properties, 4-38
reverse normals, 4-29
far fields, 6-8
calculating, 10-79
FE card, 10-71
feed, *see* sources
FEK file format, 7-1
FEK files, 4-39
FEKO, 8-1
FEKO_USER_HOME
environment variable, 2-22

FEKO_WRITE
environment variable, 2-22
FEM, 9-30
FEMAP, 9-35
FF card, 10-79
file format, 7-1
files, 10-66
CFM, 4-5, 4-46
CFX (model), 4-5
input, 2-1
output, 14-1
Parasolid, 4-21
PRE, 4-5, 4-46
summary of, 2-19
find
elements, 6-7
flares, 4-18
FM card, 9-27
FO card, 9-28
Fock area, 9-28
FOR/NEXT loops, 7-7
FP card, 9-30
FR card, 10-83, 13-1
free edges, 4-33, 4-43
free space, 4-31
frequency, 10-83
adaptive sampling, 13-1

general comments, 2-1
geometry
CAD translation, 4-22
creation, 4-16
entering, 9-33
extents, 4-16
importing, 4-22
in tree, 4-11
mirror, 4-24
modify, 4-23
point entry, 4-17
rotate, 4-24
scale, 4-24
show/hide parts, 4-10
solids, 4-18
surfaces, 4-19
translate, 4-24
validation, 4-33
wires, 4-21
geometry cards, 2-1, 9-1
**, 9-3
BL, 9-4
BP, 9-6
BQ, 9-8
BT, 9-10
CB, 9-12
CL, 9-14
CN, 9-16
DK, 9-17
DP, 9-19
DZ, 9-20
EG, 9-22
EL, 9-25
FM, 9-27
FO, 9-28
FP, 9-30
HE, 9-31
IN, 9-33
IP, 9-52
KA, 9-53
KK, 9-54
KL, 9-58
KR, 9-59
KU, 9-62
LA, 9-64
ME, 9-65
NU, 9-68
PB, 9-70
PH, 9-72
PM, 9-74
PO, 9-76
PY, 9-80
QT, 9-81
QU, 9-82
RM, 9-84
SF, 9-88
SY, 9-90
TG, 9-91
TO, 9-94
TP, 9-96
UT, 9-98
UZ, 9-100
VS, 9-101
WG, 9-104
ZY, 9-105
geometry points, imprinting, 4-28

GF card, 10-86
graph settings, 6-13
graphics cards, 4-1
Green's functions, 10-86
    showing, 6-6
grid, 4-6
ground plane, 10-54

HE card, 9-31
helices, 4-21
helix, 9-31
Hertzian electric dipole, 10-18
hide
    single item, 4-6, 4-10, 4-11
    toolbar, 4-3
hot-keys
    CADFEKO, 4-49
    EDITFEKO, 5-10

IF statement, 7-9
images, 6-16
impedance, 10-99
    loading, 10-95
    microstrip fed, 10-95
impedance boundary condition, 10-111
import
    geometry, 4-22, 4-31, 9-33
    graph data, 6-15
    meshes, 4-39, 9-33
impressed line current, 10-23, 10-29, 10-49
imprinting points, 4-28
IN card, 9-33
incident plane wave, 10-9
include files, 9-33
independent variable, 6-10
inductance
    loading, 10-94, 10-97, 10-98
input file, 2-1
intersect, 4-26
IP card, 9-52

KA card, 9-53
Kardan angles, 9-92
KK card, 9-54
KL card, 9-58
KR card, 9-59
KU card, 9-62

L4 card, 10-92
LA card, 9-64
label selected calculation, 10-100
labels, 2-8, 4-34, 9-12, 9-64
    changing, 4-41
    using, 4-41, 4-47
LD card, 10-94
LE card, 10-95
licence, 3-1
    floating, 3-2
    information, 3-1
    preferred, 3-3
    server, 3-3
line, 4-21
line segments, *see* segments
linear set of equations, 10-58
load data
    in POSTFEKO, 6-15
loaded files
    POSTFEKO, 6-17
loading, 10-118
    an edge, 10-95
    attachment point, 10-92
    distributed, 10-94
    impedance, 10-99
    microstrip line, 10-95
    parallel circuit, 10-97
    series circuit, 10-98
    showing, 6-7
local coordinates, 4-16, 4-46
local mesh parameters, 4-38
lock point entry, 4-18
loft, 4-27
losses, 10-105
LP card, 10-97
LS card, 10-98
LZ card, 10-99

machine codes, 3-4
magnetic cuboids, 9-17
magnetic dipole, 10-19
magnetic fields
    calculating, 10-71
magnetic ring current, 10-14
manual
    online, 4-49
maxalloc(m), 2-16

maximum constants, 2-16, 2-17
ME card, 9-65
medium
    dielectric, 4-31, 9-65, 9-81, 9-82
    imported models, 4-31
    magnetic, 9-81, 9-82
    properties, 4-47
memory allocation, 2-16
menu
    pop-up, 4-5, 4-13
    view, 4-4
merging
    vertices, 4-41
mesh, 4-36
    adding triangles, 4-44
    edit vertex, 4-44
    element selection, 4-36
    import, 4-39
    information, 4-40
    local settings, 4-38
    properties, 4-45
    show/hide parts, 4-10
mesh refinement, 9-84
meshing, 2-2
    non-uniform, 9-6, 9-8, 9-10
    rules, 2-5
message window, 4-3
metal
    on dielectric, 4-33
MFIE, 10-56
mirror, 4-24
MLFMM, 9-27
modelling
    rules, 2-2
models, 4-5
modify geometry, 4-23
mouse interaction, 4-5
    point entry, 4-17
multilayer substrates
    defining, 10-86
multiple reflections, 9-76
multiple selection, 4-35

N axis, 4-16
named points, 4-15
    in PRE file, 4-47
names, 4-34, 4-47
NASTRAN file, 9-39
near fields, 6-8
    calculating, 10-71
    calculation offset, 10-100
NEC file, 9-46
networks, 10-118
Neutral files, 9-35
new model, 4-5
nodes
    defining, 9-19
    definition, 2-2
    variable names, 7-10, 9-19
non-uniform mesh, 9-6, 9-8
normal vectors, 4-6, 4-27, 4-29, 9-16
    reverse, 4-29, 4-44
notes, 4-48
NU card, 9-68
NURBS surfaces, 9-68

OF card, 10-100
offset for near field calculation, 10-100
ohmic losses, 10-111
online manual, 4-49
open model, 4-4, 4-5
OPTFEKO, 11-1
    editing files, 5-9
options
    EDITFEKO, 5-5
    rendering, 4-6
orientation
    view, 6-6
OS card, 10-101
OUT file, 6-17
out-of-core, 2-16
output file, 14-1
overlap, 4-33
Overview, 1-1

pan
    2D graph, 6-14
    model view, 6-4
pan model view, 4-5
paraboloid, 4-19, 9-70
parallel execution, 5-4
parallelogram, 9-6
parameters of segmentation, 9-52
parametric, 4-13

Parasolid, 4-21
parts, 4-11
    show/hide, 4-10
PATRAN file, 9-49
PB card, 9-70
PCBMod, 10-23, 10-62
PDF viewer, 4-49
PFG files, 6-15
PFS files, 6-1
PH card, 9-72
physical optics, 9-76
planar substrate, 10-86
plane wave incidence, 10-9
plate with hole, 9-72
PM card, 9-74
PO border
    edge, 9-53
    wedge, 9-58
PO card, 9-76
PO visibility, 9-101
point entry, 4-6, 4-17
    snap, 4-8
point names, *see* nodes
points
    geometry, 4-9
    imprinting, 4-28
    in PRE file, 4-47
    named, 4-15
Polar plots, 6-13
polarisation, 6-16
polygons, 4-19, 9-80
    definition, 2-3
    meshed, 9-74
    normals, 4-44
    UTD formulation, 9-98
polyline, 4-21
port, 3-2
POSTFEKO, 6-1
    command line, 6-1
    control toolbar, 6-2
    graphs, 6-1
    help, 6-18
    load/import data, 6-15
    loading sessions, 6-1
    overview, 6-2
    sessions, 6-1
power input, 10-105
PRE files, 4-5, 4-46
    named points, 4-47
    variables, 4-47
PREFEKO, 7-1
preferences
    CADFEKO, 4-46
    EDITFEKO, 5-5
preferred licence, 3-3
PRINT command, 7-11
priority
    setting, 8-1
program execution control, 10-103
project notes, 4-48
projection, 4-28
properties
    medium, 4-31
    mesh, 4-38, 4-45
    modify, 4-23
PS card, 10-103
PW card, 10-105
PY card, 9-80

QT card, 9-81
QU card, 9-82
quadrangle, 9-8

radiation
    patterns, 10-79
    patterns as sources, 10-38
radius
    segments, 4-36, 4-38
raw data, 6-15
    series, 6-12, 6-14
real ground, 10-54, 10-86
redo
    model, 4-4
    selection, 4-36
    view operations, 4-9
referencing elements, 4-47
reflectors, 4-19
regions, 4-11
    properties, 4-31, 4-38
remeshing, 9-84
remote execution, 5-4
remove
    duplicates, 4-42
    slivers, 4-42

renaming
geometry, 4-23
mesh elements, 4-41
rendering, 4-1
options, 4-6
requested field points, 6-6
resistance loading, 10-94, 10-95, 10-97, 10-98
restore
deleted faces / edges, 4-31
results
checking validity, 2-26
reuse objects, 4-30
RM card, 9-84
rotate, 6-4
geometry, 4-24, 9-91
model view, 4-5
parts, 4-26
plots, 6-13
point, 9-96
workplane, 4-8
ruled surface, 4-27
run, 4-46
from GUI, 5-4

S-parameters, 10-117
SA card, 10-109
SAR, 6-9
specific absorption rate, 10-109
save
as UNIX, 5-3
model, 4-4, 4-5
scale, 4-24
geometry, 9-88, 9-91
Parasolid, 4-21
units, 4-47
search for elements, 6-7
secfeko, 3-1
segmentation
parameters, 9-52
rules for, 2-5
segments, 9-4
arc, 9-14
coating, 10-63
creation, *see* geometry cards
definition, 2-2
helix, 9-31
radius, 4-36, 4-38
selection
and cutplanes, 4-35
edges, 4-36
in 3D view, 4-35
items, 4-34
mesh elements, 4-36
multiple, 4-35
toolbar, 4-35
series
adding, 6-12
calculations, 6-12
server
licence, 3-2
sessions
POSTFEKO, 6-1
SF card, 9-88
shells, 4-31
short-cut keys, 4-49
show
axes, 6-6
calculation points, 6-6
excitations, 6-7
Green's functions, 6-6
loads, 6-7
single item, 4-6, 4-10, 4-11
toolbar, 4-3
transmission lines, 6-7
sides
triangles, 4-44
simplify, 4-29
SK card, 10-111
skin effect, 10-111
slivers
removing, 4-42
snap, 4-8
solid bodies, 4-31
solid geometry, 4-18
solids and shells, 4-31, 4-47
solution, 5-4
options, 4-46
solving, 4-46
sources, 10-6
Hertzian electric dipole, 10-18
impressed line current, 10-23, 10-29, 10-49
incident plane wave, 10-9
magnetic dipole, 10-19
magnetic ring current, 10-14

microstrip line, 10-26
patch feed pin, 10-16
radiation patterns, 10-38
showing, 6-7
spherical modes, 10-43
voltage on a node, 10-13
voltage on a segment, 10-12
voltage on an edge, 10-21, 10-26
waveguide modes, 10-51
SP card, 10-117
specified vertex, 4-38
spheres, 4-18, 9-62
dielectric/magnetic, 9-17
spherical modes, 10-43
spinning parts, 4-26
spiral, 9-31
spirals, 4-21
splines, 4-21
split parts, 4-26
status bar, 4-3
STL file, 9-48
stripline
feeding, 10-26
loading, 10-92
substrate, 10-86
subtract, 4-26
superuser mode
in EDITFEKO, 5-6
surface currents
distribution, 10-101
surface triangles, *see* elements
surfaces, 4-19, 4-31
creating solids, 4-26
metal, 4-33
suspect, 4-23
sweeping parts, 4-26
SY card, 9-90
symbolic variables, 7-2
symmetry, 2-10, 4-48, 9-90
electric, 2-11
geometric, 2-10
magnetic, 2-11

tetrahedral volume elements, 9-81
tetrahedron
definition, 2-3
TG card, 9-91
thin dielectric sheet, 10-111
TIMEFEKO, 12-1
TL card, 10-118
TO card, 9-94
tolerance, 4-41
toolbars
3D view, 4-9
POSTFEKO, 6-2
selection, 4-3, 4-35
showing, 4-3
standard, 4-4
toroidal segment, 9-94
TP card, 9-96
transformation, 4-24, 6-6, 9-91
of point, 9-96
workplane, 4-8
translation, 4-24, 9-91, 9-96
transmission lines, 10-118
coupling, 10-23, 10-62
showing, 6-7
tree, 4-11
showing, 4-3
triangles, *see* elements
back side, 4-44
creating, 4-44
normal side, 4-44
normals, 4-44
remove duplicates, 4-42

U axis, 4-6, 4-16
unconnected edges, 4-33, 4-43
undo
list depth, 4-46
meshing, 4-4
model, 4-4
selection, 4-36
view operations, 4-9
union, 4-26
units, 4-47
updates, 5-5
UT card, 9-98
UTD, 6-9, 9-98
cylinder, 9-100
polygon, 9-80
UZ card, 9-100

V axis, 4-6, 4-16

validation
geometry, 4-33
value series, 6-12, 6-14
variables, 4-13, 4-15
environment, 2-22
in point names, 7-10
in PRE file, 4-47
memory allocation, 2-16
predefined, 2-16
symbolic, 7-2
version, 4-49
vertices
editing, 4-44
merging, 4-41
specified location, 4-38
view
orientation, 6-6
out file, 6-17
view selected edges, 4-36
views, 4-4
manipulation, 4-5
visibility
PO region, 9-101
view items, 4-9
voltage source
on a node, 10-13
on a segment, 10-12
on edge, 10-21, 10-26
VS card, 9-101

waveguide port, 10-51
wedge as PO border, 9-58
WFG files, 6-15
WG card, 9-104
what's this, 4-5
wire
curved, 9-14
objects, 4-21
segments, 4-36
wire grid, 9-104
wire segments, *see* segments
workplane, 4-6
fit to geometry, 4-7
transformation, 4-8

zoom, 6-4
2D graph, 6-14
zoom model view, 4-5
ZY card, 9-105